US007620844B2

(12) **United States Patent**
Mambakkam et al.

(10) **Patent No.: US 7,620,844 B2**
(45) **Date of Patent: *Nov. 17, 2009**

(54) **FIELD-OPERABLE, STAND-ALONE APPARATUS FOR MEDIA RECOVERY AND REGENERATION**

(75) Inventors: **Sreenath Mambakkam**, San Jose, CA (US); **Larry Lawson Jones**, Palo Alto, CA (US); **Arockiyaswamy Venkidu**, Menlo Park, CA (US)

(73) Assignee: **MCM Portfolio LLC**, Cupertino, CA (US)

(*) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 0 days.

This patent is subject to a terminal disclaimer.

(21) Appl. No.: **11/844,330**

(22) Filed: **Aug. 23, 2007**

(65) **Prior Publication Data**

US 2007/0288677 A1 Dec. 13, 2007

Related U.S. Application Data

(63) Continuation of application No. 11/075,496, filed on Mar. 8, 2005, now Pat. No. 7,278,051, which is a continuation-in-part of application No. 10/253,547, filed on Sep. 23, 2002, now Pat. No. 6,839,864, which is a continuation-in-part of application No. 10/064,967, filed on Sep. 4, 2002, now Pat. No. 7,093,161, and a continuation-in-part of application No. 10/167,925, filed on Jun. 11, 2002, now Pat. No. 7,222,205, which is a continuation of application No. 09/610,904, filed on Jul. 6, 2000, now Pat. No. 6,438,638.

(51) **Int. Cl.**
G06F 11/00 (2006.01)

(52) **U.S. Cl.** **714/6**

(58) **Field of Classification Search** 714/6, 714/2, 7, 8, 13, 15, 37, 42

See application file for complete search history.

(56) **References Cited**

U.S. PATENT DOCUMENTS

4,092,732	A	5/1978	Ouchi
5,296,692	A	3/1994	Shino
5,394,206	A	2/1995	Cocca
5,396,617	A	3/1995	Villwock et al.
5,436,621	A	7/1995	Macko et al.
5,437,020	A	7/1995	Wells et al.
5,471,038	A	11/1995	Eisele et al.
5,485,606	A	1/1996	Midgdey
5,497,464	A	3/1996	Yeh
5,522,049	A	5/1996	Kimura et al.
5,538,436	A	7/1996	Garney
5,576,698	A	11/1996	Card et al.

(Continued)

FOREIGN PATENT DOCUMENTS

DE	20109810	8/2001

(Continued)

OTHER PUBLICATIONS

Actiontec, "CameraConnect Pro Parallel Port Flash Card Reader User's Manual," available at least by Oct. 28, 1999.

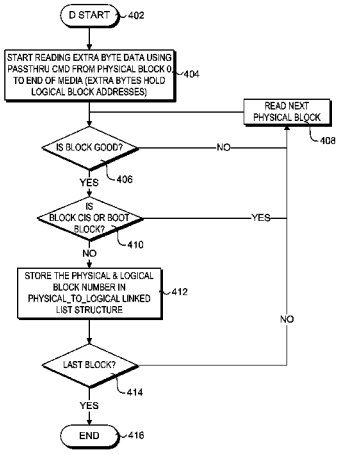
(Continued)

Primary Examiner—Nadeem Iqbal
(74) *Attorney, Agent, or Firm*—Greenberg Traurig, LLP

(57) **ABSTRACT**

A method for recovering data from corrupted flash media and regenerated damaged flash media have been disclosed. In one embodiment, the method comprises accessing data on a flash media, searching the data for file indicia corresponding at least one predetermined file type, reading the data based on information in the file indicia, the data including a recovered file, and storing the recovered file on a storage medium. Other embodiments have been claimed and described.

32 Claims, 22 Drawing Sheets



U.S. PATENT DOCUMENTS

5,584,043	A		12/1996	Burkart
5,589,719	A		12/1996	Fiset
5,596,562	A		1/1997	Chen
5,604,917	A		2/1997	Saito et al.
5,630,174	A		5/1997	Stone et al.
5,640,541	A		6/1997	Bartram et al.
5,679,007	A		10/1997	Potdevin et al.
5,708,799	A		1/1998	Gafken et al.
5,717,951	A		2/1998	Yabumoto
5,729,204	A		3/1998	Fackler et al.
5,734,894	A		3/1998	Adamson et al.
5,740,349	A		4/1998	Hasbun et al.
5,752,857	A		5/1998	Knights
5,786,769	A		7/1998	Coteus et al.
5,790,878	A		8/1998	Anderson et al.
5,799,200	A	*	8/1998	Brant et al. 713/340
5,802,553	A		9/1998	Robinson et al.
5,805,834	A		9/1998	McKinley et al.
5,815,426	A		9/1998	Jigour et al.
5,828,905	A		10/1998	Rao
5,839,108	A		11/1998	Daberko
5,844,910	A		12/1998	Niijima et al.
5,844,911	A		12/1998	Schadegg et al.
5,877,975	A		3/1999	Jigour et al.
5,887,145	A		3/1999	Harari et al.
5,892,213	A		4/1999	Ito et al.
5,905,888	A		5/1999	Jones et al.
5,928,347	A		7/1999	Jones
5,928,370	A		7/1999	Asnaashari
5,929,416	A		7/1999	Dos Santos Pato et al.
5,930,496	A		7/1999	MacLaren et al.
5,933,328	A		8/1999	Wallace et al.
5,956,473	A	*	9/1999	Ma et al. 714/5
5,961,652	A		10/1999	Thompson
5,964,885	A		10/1999	Little et al.
5,974,426	A	*	10/1999	Lee et al. 707/202
5,995,376	A		11/1999	Schultz et al.
6,002,605	A		12/1999	Iwasaki et al.
6,006,295	A		12/1999	Jones et al.
6,009,492	A		12/1999	Matsuoka
6,010,066	A		1/2000	Itou et al.
6,011,741	A		1/2000	Wallace et al.
6,015,093	A		1/2000	Barrett et al.
6,026,007	A		2/2000	Jigour et al.
6,038,400	A		3/2000	Bell et al.
6,061,746	A		5/2000	Stanley et al.
6,062,887	A		5/2000	Schuster et al.
6,067,234	A		5/2000	Kim et al.
6,075,706	A		6/2000	Learmonth et al.
6,079,621	A		6/2000	Vardanyan et al.
6,088,755	A		7/2000	Kobayashi et al.
6,088,802	A		7/2000	Bialick et al.
6,097,605	A		8/2000	Klatt et al.
6,102,715	A		8/2000	Centofante
6,112,014	A		8/2000	Kane
6,132,223	A		10/2000	Seeley et al.
6,137,710	A		10/2000	Iwasaki et al.
6,145,046	A		11/2000	Jones
6,170,029	B1		1/2001	Kelley et al.
6,170,066	B1	*	1/2001	See 714/22
6,173,291	B1		1/2001	Jenevein
6,175,517	B1		1/2001	Jigour et al.
6,182,162	B1		1/2001	Estakhri et al.
6,189,055	B1		2/2001	Eisele et al.
6,199,122	B1		3/2001	Kobayashi
6,202,932	B1		3/2001	Rapeli
6,203,378	B1		3/2001	Shobara et al.
6,226,202	B1		5/2001	Kikuchi
6,234,844	B1		5/2001	Somerville et al.
6,264,506	B1		7/2001	Yasufuku et al.
6,266,724	B1		7/2001	Harari et al.
6,279,061	B1		8/2001	Aoki et al.
6,279,069	B1		8/2001	Robinson et al.
6,282,612	B1		8/2001	Sakajiri et al.
6,292,863	B1		9/2001	Terasaki et al.
6,317,352	B1		11/2001	Halbert et al.
6,330,688	B1		12/2001	Brown
6,353,870	B1		3/2002	Mills et al.
6,378,015	B1		4/2002	Yen
6,381,662	B1		4/2002	Harari et al.
6,385,677	B1		5/2002	Yao
6,386,920	B1		5/2002	Sun
6,393,524	B1		5/2002	Ayers
6,402,558	B1		6/2002	Hung-Ju et al.
6,405,323	B1		6/2002	Lin et al.
6,408,352	B1		6/2002	Hosaka et al.
6,413,108	B2		7/2002	Centofante
6,426,801	B1		7/2002	Reed
6,468,101	B2		10/2002	Suzuki
6,482,029	B2		11/2002	Nishimura
6,490,163	B1		12/2002	Pua et al.
6,581,830	B1		6/2003	Jelinek et al.
6,595,803	B2		7/2003	Akagi et al.
6,599,147	B1		7/2003	Mills et al.
6,601,124	B1		7/2003	Blair
6,607,405	B2		8/2003	Nishimura
6,663,007	B1		12/2003	Sun
6,666,724	B1		12/2003	Lwee
6,675,233	B1		1/2004	Du et al.
6,684,283	B1		1/2004	Harris et al.
6,705,529	B1		3/2004	Kettunen et al.
6,738,259	B2		5/2004	Le et al.
6,745,267	B2		6/2004	Chen et al.
6,746,280	B1		6/2004	Lu et al.
6,761,313	B2		7/2004	Hsieh et al.
6,761,320	B1		7/2004	Chen
6,808,424	B2		10/2004	Kaneshiro et al.
6,839,864	B2	*	1/2005	Mambakkam et al. 714/5
6,859,369	B2		2/2005	Mambakkam et al.
6,973,535	B2		12/2005	Bruner et al.
6,984,152	B2		1/2006	Mowery et al.
7,093,161	B1	*	8/2006	Mambakkam et al. 714/15
7,095,618	B1		8/2006	Mambakkam et al.
7,162,547	B2		1/2007	Hosaka et al.
7,191,270	B2		3/2007	Oh et al.
7,252,240	B1		8/2007	Jones et al.
7,295,443	B2		11/2007	Mambakkam et al.
7,412,552	B2		8/2008	Jones et al.
2001/0039603	A1		11/2001	Manowitz
2002/0069363	A1		6/2002	Winburn
2002/0178307	A1		11/2002	Pua et al.
2002/0185533	A1		12/2002	Shieh et al.
2003/0038177	A1		2/2003	Morrow
2003/0041203	A1		2/2003	Jones et al.
2003/0041284	A1	*	2/2003	Mambakkam et al. 714/15
2003/0070112	A1		4/2003	York
2003/0074529	A1		4/2003	Crohas
2003/0084220	A1		5/2003	Jones et al.
2003/0093606	A1		5/2003	Mambakkam et al.
2003/0172263	A1		9/2003	Liu
2004/0027879	A1		2/2004	Chang
2005/0273648	A1		12/2005	Mambakkam et al.
2006/0059385	A1		3/2006	Atri et al.
2006/0242460	A1		10/2006	Mambakkam et al.
2006/0253636	A1		11/2006	Jones et al.
2007/0180177	A1		8/2007	Jones et al.
2008/0009196	A1		1/2008	Mambakkam et al.
2008/0017718	A1		1/2008	Jones et al.
2008/0250174	A1		10/2008	Jones et al.

2008/0299809	A1	12/2008	Mambakkam et al.
2009/0100207	A1	4/2009	Jones et al.

FOREIGN PATENT DOCUMENTS

EP	0775964	9/1996
EP	0987876	8/1997
EP	1043884	10/2000
EP	1139208	10/2001
JP	6195524	7/1994
JP	08235028	9/1996
JP	10154211	6/1998
JP	1115928	1/1999
JP	11053485	2/1999
JP	2000286564	10/2000
JP	2002157056	5/2002
TW	490889	6/2002
WO	0023936	4/2000
WO	2004027617	4/2004

OTHER PUBLICATIONS

Actiontec, "CameraConnect Pro," available at least by Oct. 28, 1999.

Burge, Legand L., et al., "A Ubiquitous Stable Storage for Mobile Computing Devices," ACM, Proceedings of the 2001 ACM Symposium on Applied Computing, pp. 401-404, Mar. 2001.

DataFab Systems, Inc., "DataFab Systems Inc., Leading in Portable Storage Systems, Is Now Offering Dual-Slot CompactFlash and SmartMedia Card Reader," Sep. 10, 1999.

Jones, Larry Lawson et al., U.S. Appl. No. 11/003,185 entitled "Flashtoaster for Reading Several Types of Flash Memory Cards with or without a PC," filed Dec. 2, 2004.

Lexmark International, Inc., Service Manual for 5000 and 5700 Color Jetprinter & 5770 Photo Jetprinter, Oct. 2000.

Microtech International, Inc., "Microtech Delivers Industry's First 3 Slot SCSI Digital Film Reader," Jan. 5, 2000.

Microtech International, Inc., "Microtech Digital Photography Solutions," available at least by Feb. 26, 2000.

Microtech International, Inc., "Microtech PCD-47B SCSI Digital Film Reader/Writer," available at least by May 24, 2000.

Microtech International, Inc., "Microtech USB CameraMate Supports IBM Microdrive," Feb. 18, 1999.

Microtech International, Inc., "PCD-47 User's Manual," Version 1.1, available at least by May 24, 2000.

Steve's Digicams, "CardMate PCF-100 User Review," Apr. 12, 1999.

Steve's Digicams, "Microtech USB CameraMate User Review," Sep. 5, 1999.

USPTO Transaction History of U.S. Appl. No. 10/253,547, filed Sep. 23, 2002, entitled "Field Operable Stand Alone Apparatus for Media Recovery and Regeneration," now U.S. Patent No. 6,839,864.

USPTO Transaction History of U.S. Appl. No. 11/075,496, filed Mar. 8, 2005, entitled "Field Operable Stand Alone Apparatus for Media Recovery and Regeneration," now U.S. Patent No. 7,278,051.

Antec, Inc., "PhotoChute3 USB," product manual, pp. 1-18, available at least by Apr. 26, 1999.

CompactFlash Association, "CF+ and Compact Flash Specification," Rev. 1.4, Jul. 1999.

CQ Publishing of Japan, "Interface," pp. 52-131, Dec. 1, 1999 (article and English translation).

DataRescue sa/nv, Inc., "PhotoRescue User's Guide," rev. 1.0, pp. 1-8, 2001.

DataRescue sa/nv, Inc., DataRescue Home Page, located at http://web.archive.org/web/20010722191109/http://datarescue.com, archived Jul. 22, 2001.

DataRescue sa/nv, Inc., DataRescue PhotoRescue™ Specifications, located at http://web.archive.org/web/20010827073251/www.datarescue.com/photorescue/spec.htm, archived Aug. 27, 2001.

Galbraith, Rob, "Building the Ultimate Photo Recovery Kit," located at http://www.robgalbraith.com/bins/content_page.asp?cid=7-4419-4501, Jan. 23, 2002.

Lexar Media Web Pages—Parallel Port, Universal Readers, FAQ, Jumpshot, Jun. 5, 2000.

Microtouch Smart Media to PCMCIA Adapter Product Sheet, Jun. 8, 2000.

Ontrack Data International, Inc., "EasyRecovery™ Professional Edition User Guide," pp. 1-45, 2000.

Supplementary Search Report for EP Application No. 01952974.2, Sep. 21, 2004.

Twice.com, "Digital Imaging Well Exposed at RetailVision," available at least by Apr. 26, 1999.

* cited by examiner

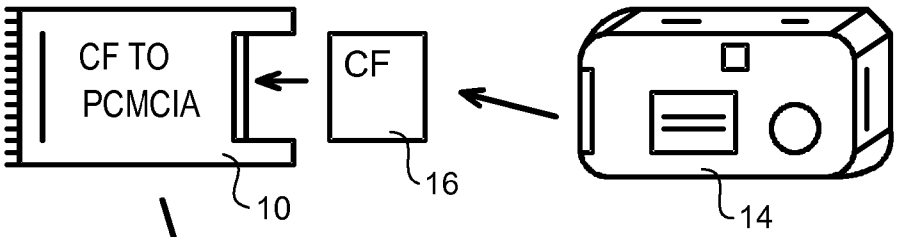


Fig. 1A

PRIOR ART

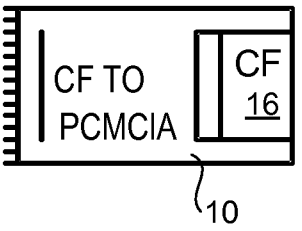
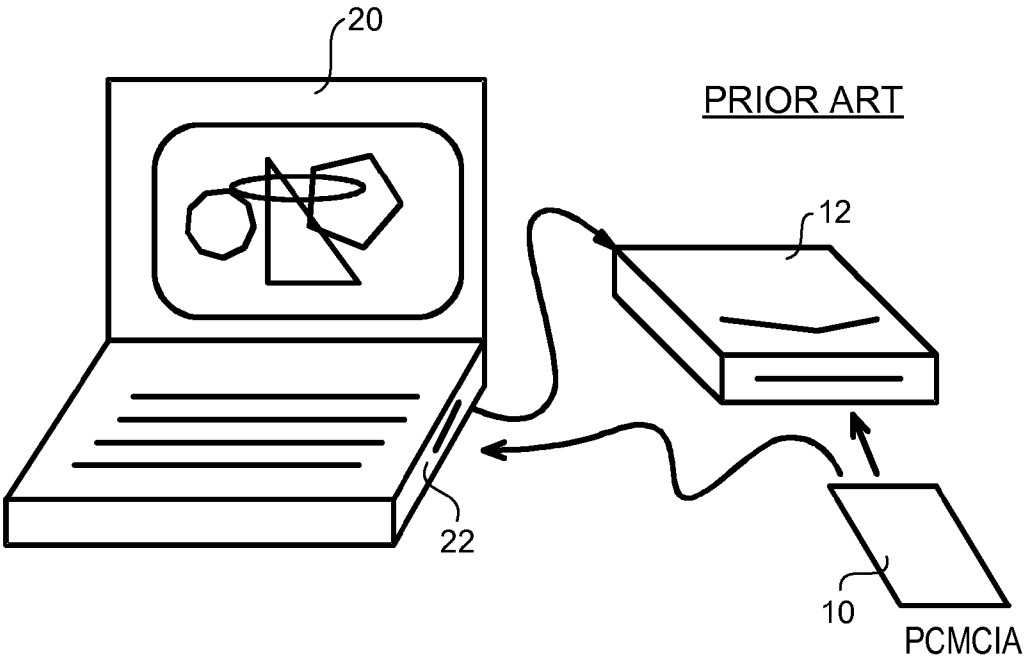


Fig. 1B



PRIOR ART

Fig. 1C

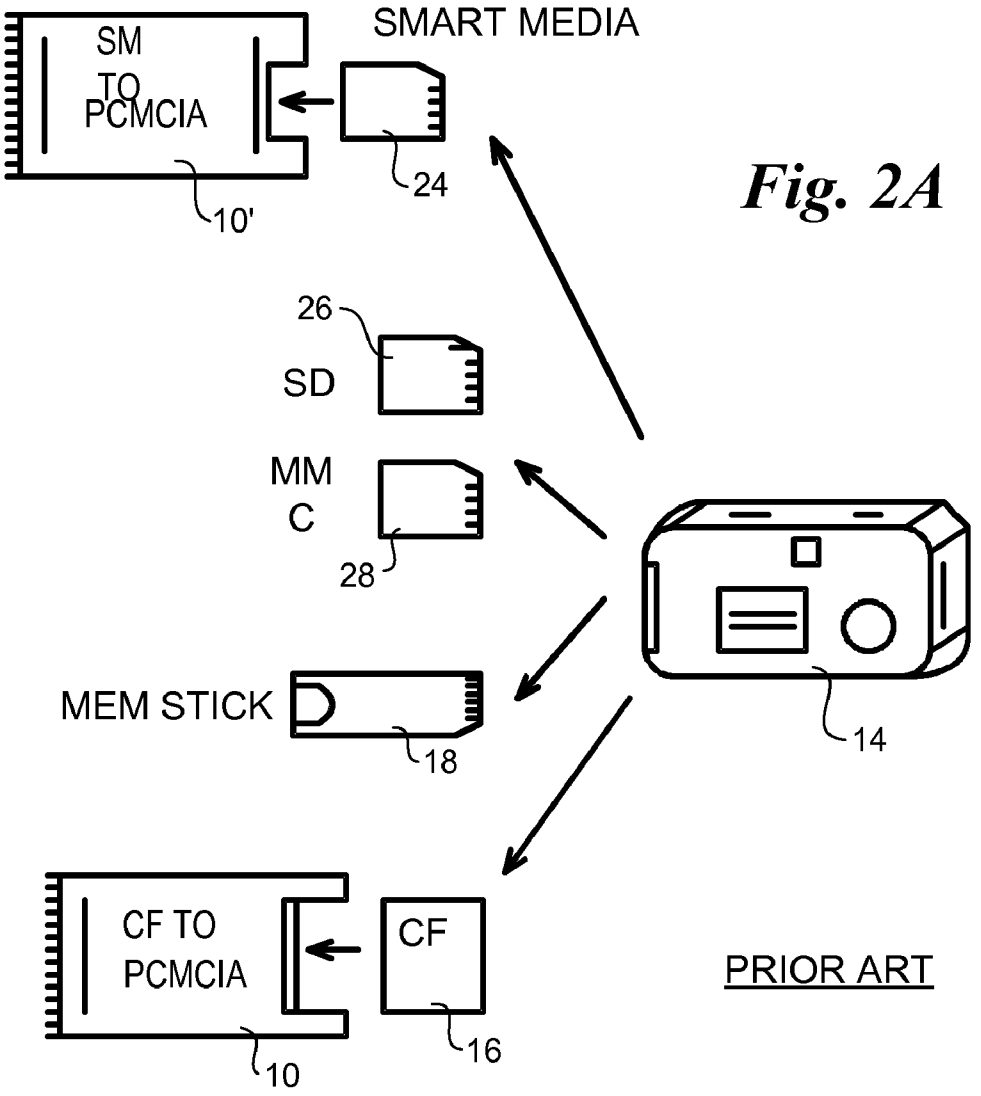

SM TO PCMCIA
SMART MEDIA
24
10'
26
SD
MM C
28
MEM STICK
18
14
CF TO PCMCIA
CF
16
10
PRIOR ART

Fig. 2A

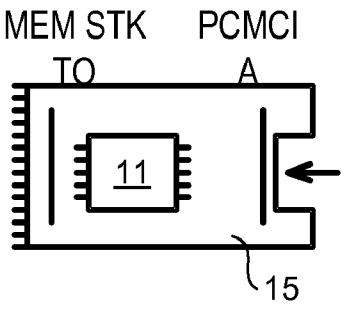

MEM STK TO
PCMCI A
11
15

Fig. 2B

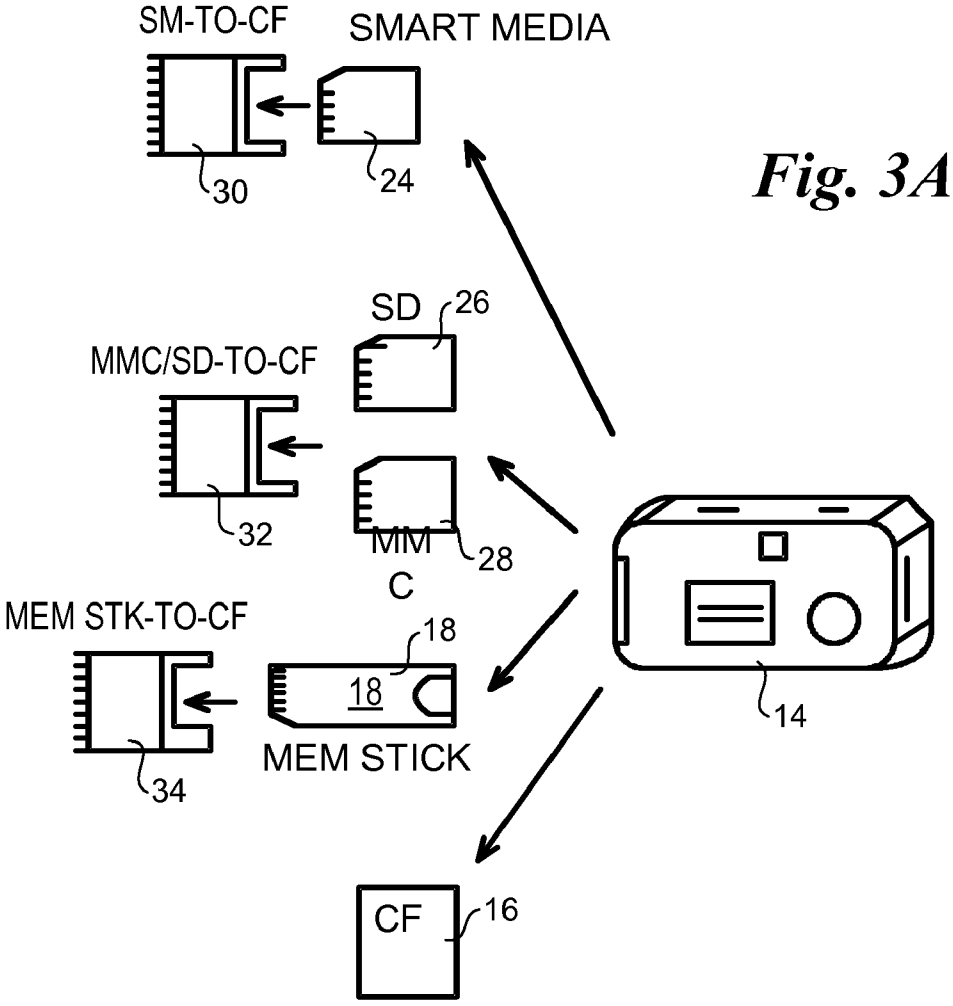

SM-TO-CF
SMART MEDIA
30
24
SD
26
MMC/SD-TO-CF
32
MM
C
28
MEM STK-TO-CF
18
18
MEM STICK
34
14
CF
16

Fig. 3A

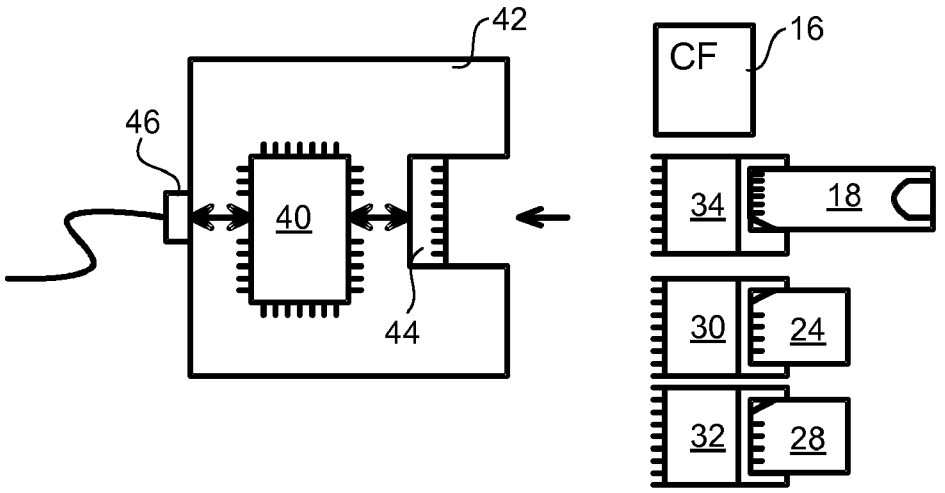

42
46
40
44
CF
16
34
18
30
24
32
28

Fig. 3B

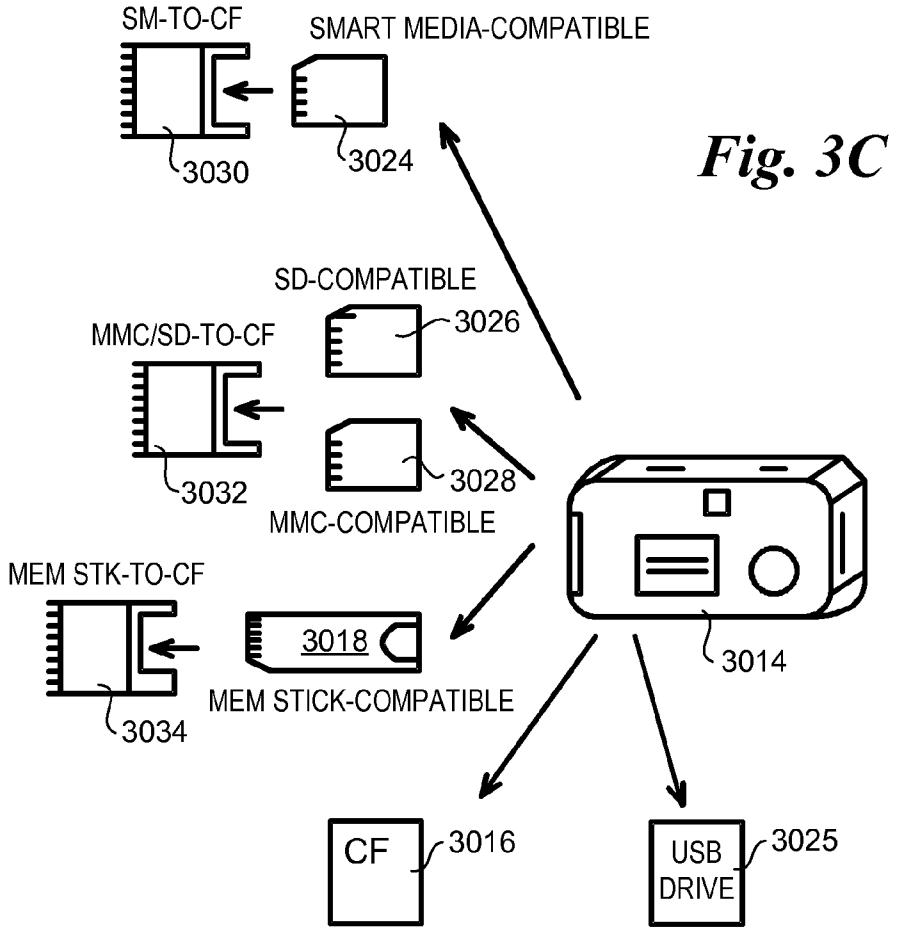

SM-TO-CF
SMART MEDIA-COMPATIBLE
3030
3024
SD-COMPATIBLE
3026
MMC/SD-TO-CF
3032
3028
MMC-COMPATIBLE
MEM STK-TO-CF
3018
MEM STICK-COMPATIBLE
3034
3014
CF
3016
USB DRIVE
3025

Fig. 3C

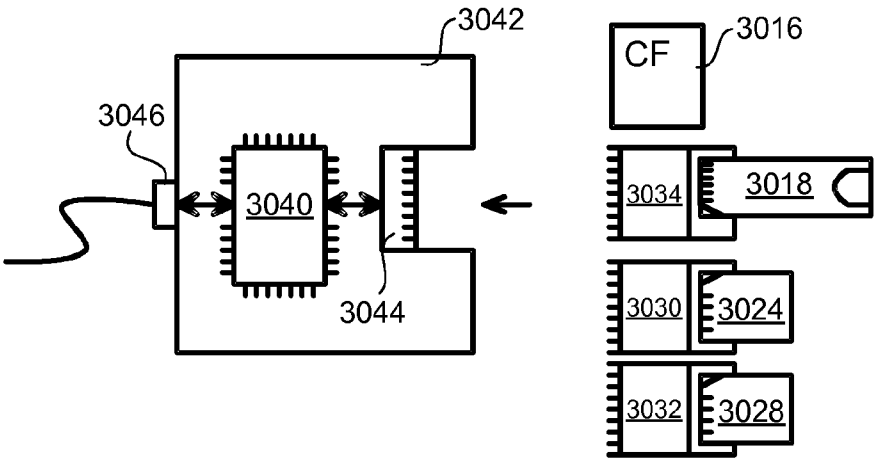

3042
3046
3040
3044
CF
3016
3034
3018
3030
3024
3032
3028

Fig. 3D

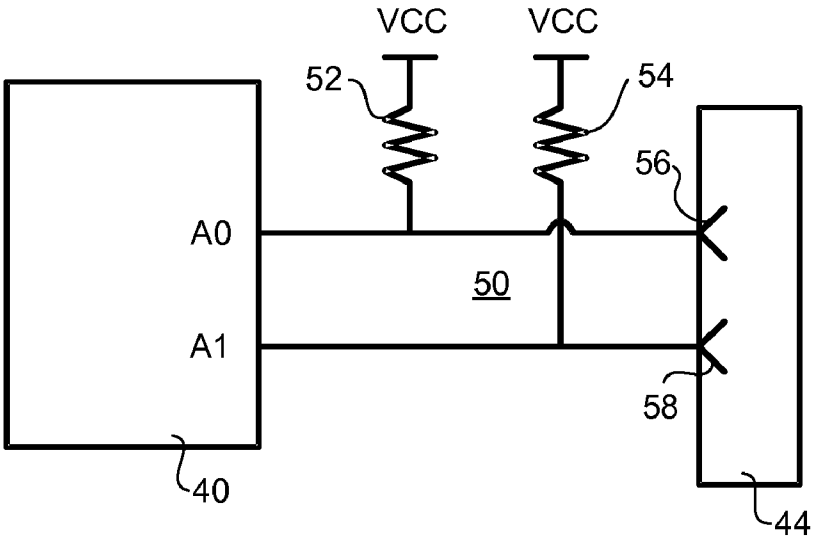

VCC
VCC
52
54
56
A0
50
A1
58
40
44

Fig. 4A

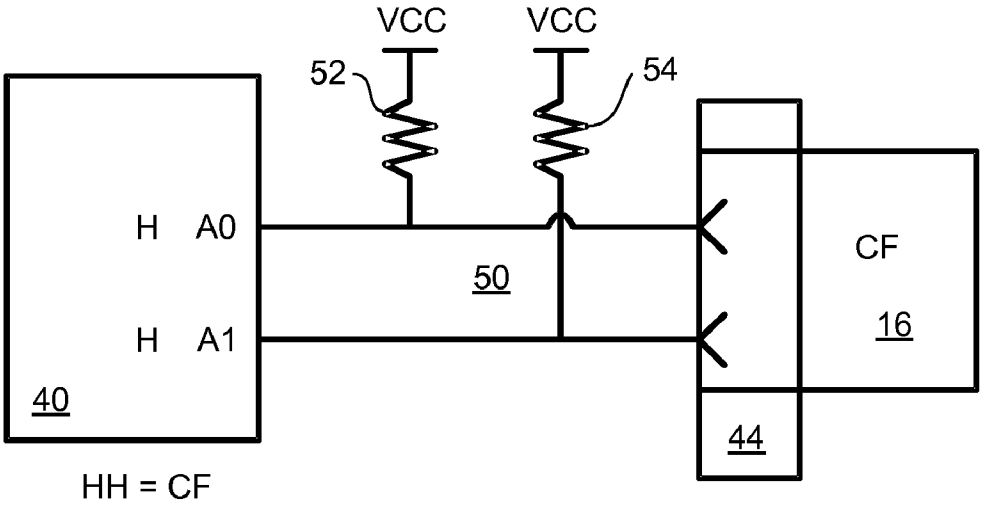
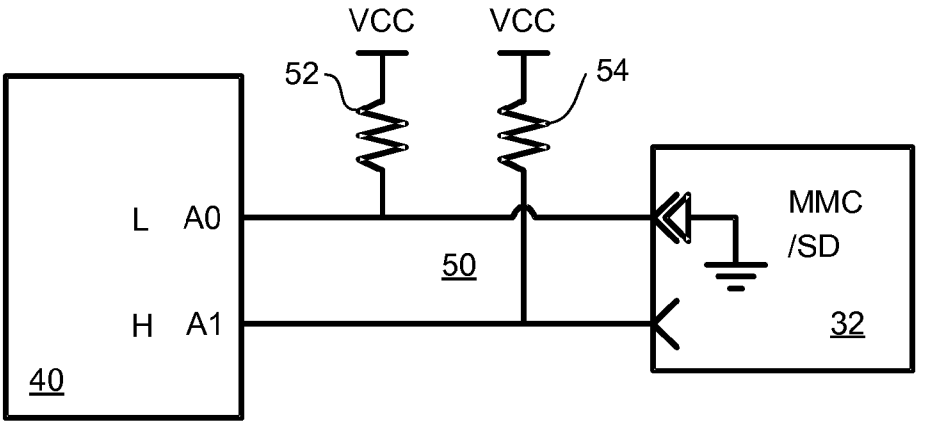
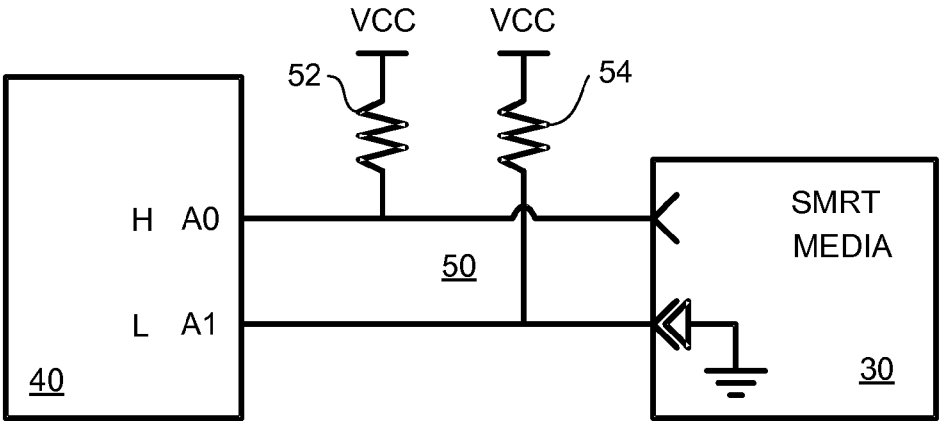

VCC
VCC
52
54
H A0
50
H A1
CF
16
40
44
HH = CF

Fig. 4B



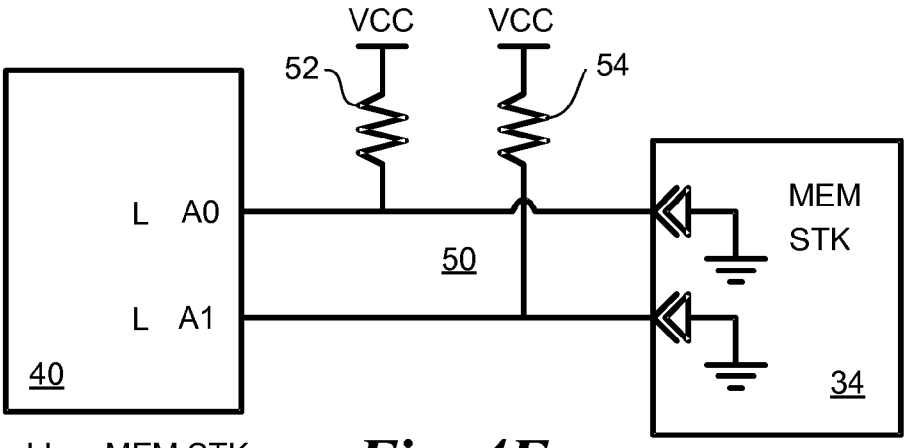
LH = MMC/SD

Fig. 4C



HL = SMRT MEDIA

Fig. 4D



LL = MEM STK

Fig. 4E

Pin	CF	Smart Media	MMC/SD	Memory Stick
1	Ground	Ground	Ground	Ground
2	D3	D3	---	---
3	D4	D4	---	---
4	D5	D5	---	---
5	D6	D6	---	---
6	D7	D7	---	---
7	-CE1	-SMCS	---	---
8	A10	---	---	---
9	-OE	-OE	---	---
10	A9	---	---	---
11	A8	---	---	---
12	A7	---	---	---
13	Power	Power	Power	Power
14	A6	CLE		
15	A5	ALE		
16	A4	READY		
17	A3	-WP		
18	A2	LVD	SCLK	SCLK
19	A1		DIO	DIO
20	A0		CMD	BS
21	D0	D0		
22	D1	D1		
23	D2	D2		
24	---	---	---	---
25	-CD2	-CD2	-CD2	-CD2
26	-CD1	-CD1	-CD1	-CD1
27	D11	---	---	---
28	D12	---	---	---
29	D13	---	---	---
30	D14	---	---	---
31	D15	---	---	---
32	-CE2	---	---	---
33	---	---	---	---
34	tie high	---	---	---
35	tie high	---	---	---
36	-WE	-WE	---	---
37	INTRQ	---	---	---
38	Power	Power	Power	Power
39	---	---	---	---
40	---	---	---	---
41	RESET	---	---	---
42	---	---	---	---
43	---	---	---	---
44	-REG	---	---	---
45	---	---	---	---
46	---	---	---	---
47	D8	---	---	---
48	D9	---	---	---
49	D10	---	---	---
50	Ground	Ground	Ground	Ground

Fig. 5

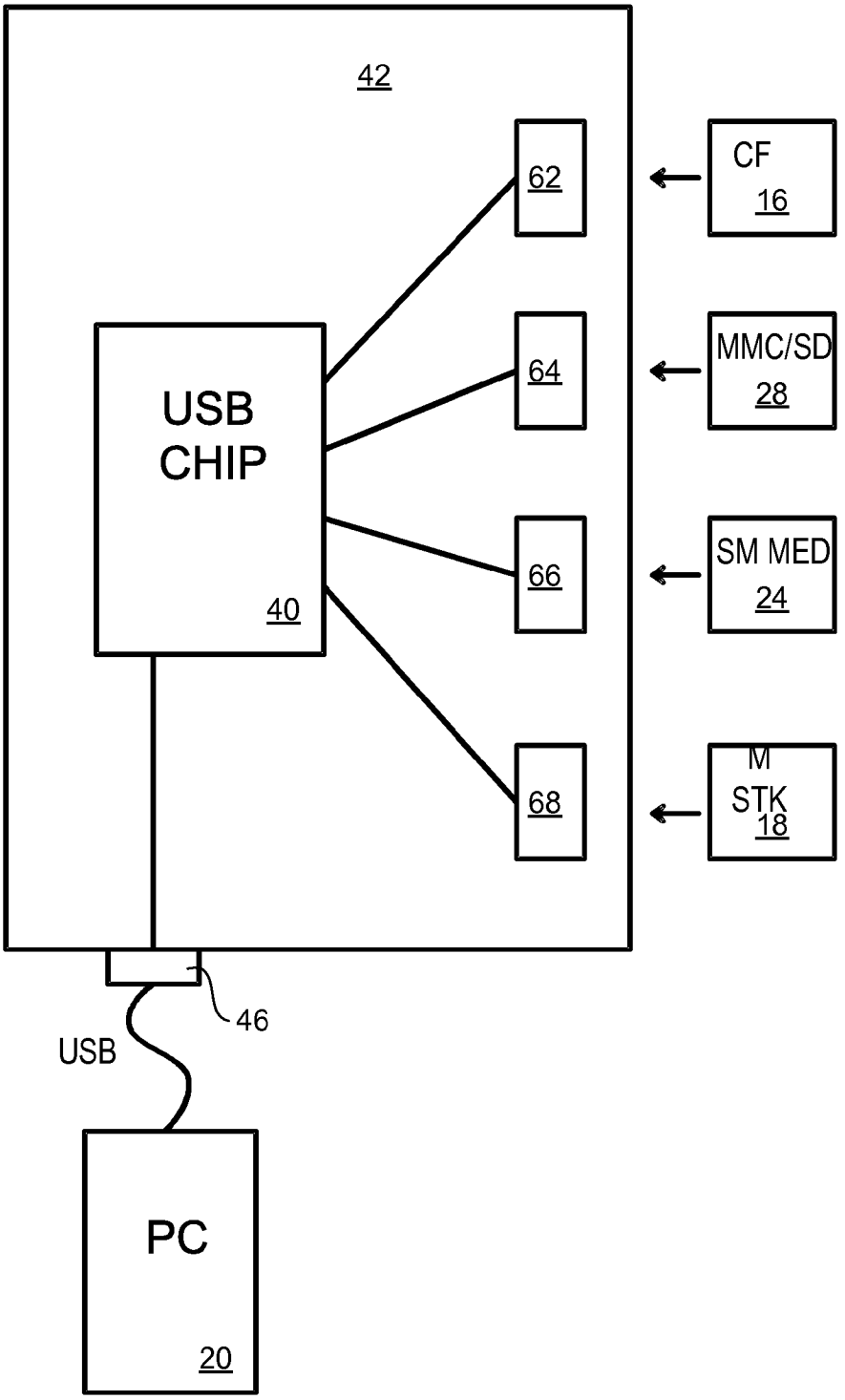

42
62
64
66
68
CF
16
MMC/SD
28
SM MED
24
M
STK
18
USB
CHIP
40
46
USB
PC
20

Fig. 6

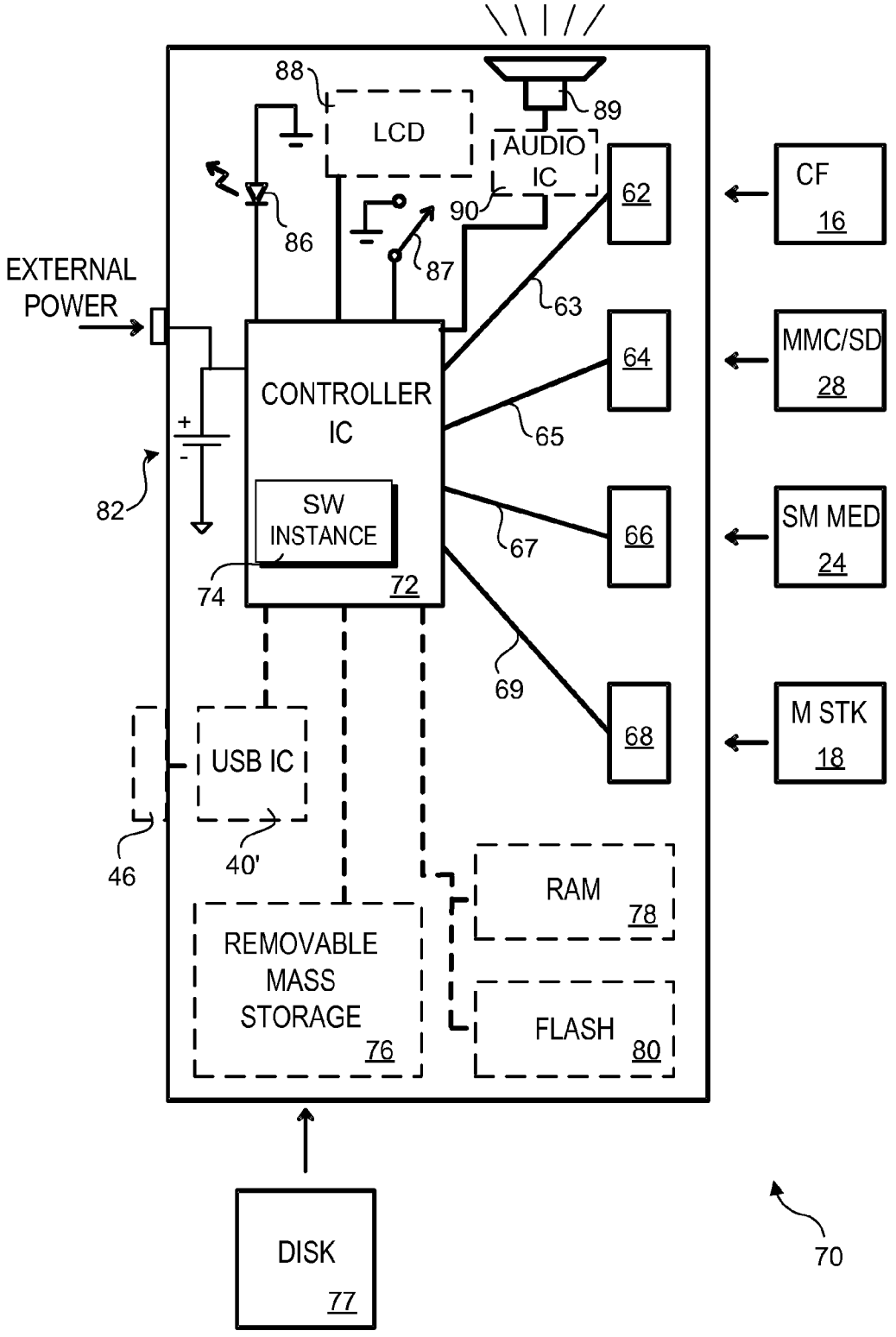

88
LCD
89
AUDIO IC
90
86
87
62
CF 16
EXTERNAL POWER
63
64
MMC/SD 28
CONTROLLER IC
65
82
SW INSTANCE
74
72
67
66
SM MED 24
69
68
M STK 18
USB IC
46
40'
RAM 78
REMOVABLE MASS STORAGE 76
FLASH 80
DISK 77
70

Fig. 7

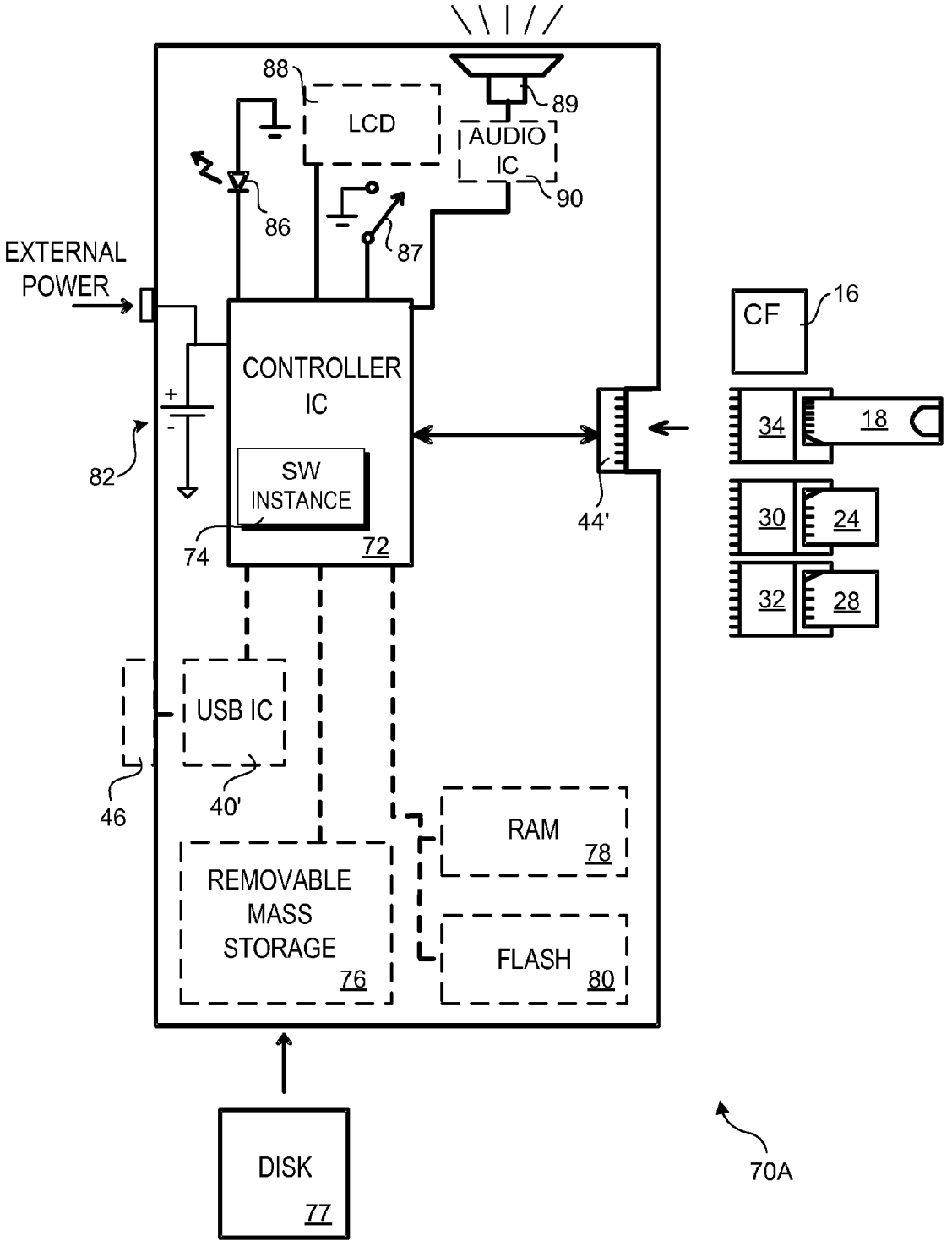

88
LCD
89
AUDIO IC
90
86
87
EXTERNAL POWER
CONTROLLER IC
SW INSTANCE
72
74
82
44'
CF
16
34
18
30
24
32
28
USB IC
40'
46
RAM
78
REMOVABLE MASS STORAGE
76
FLASH
80
DISK
77
70A

Fig. 8

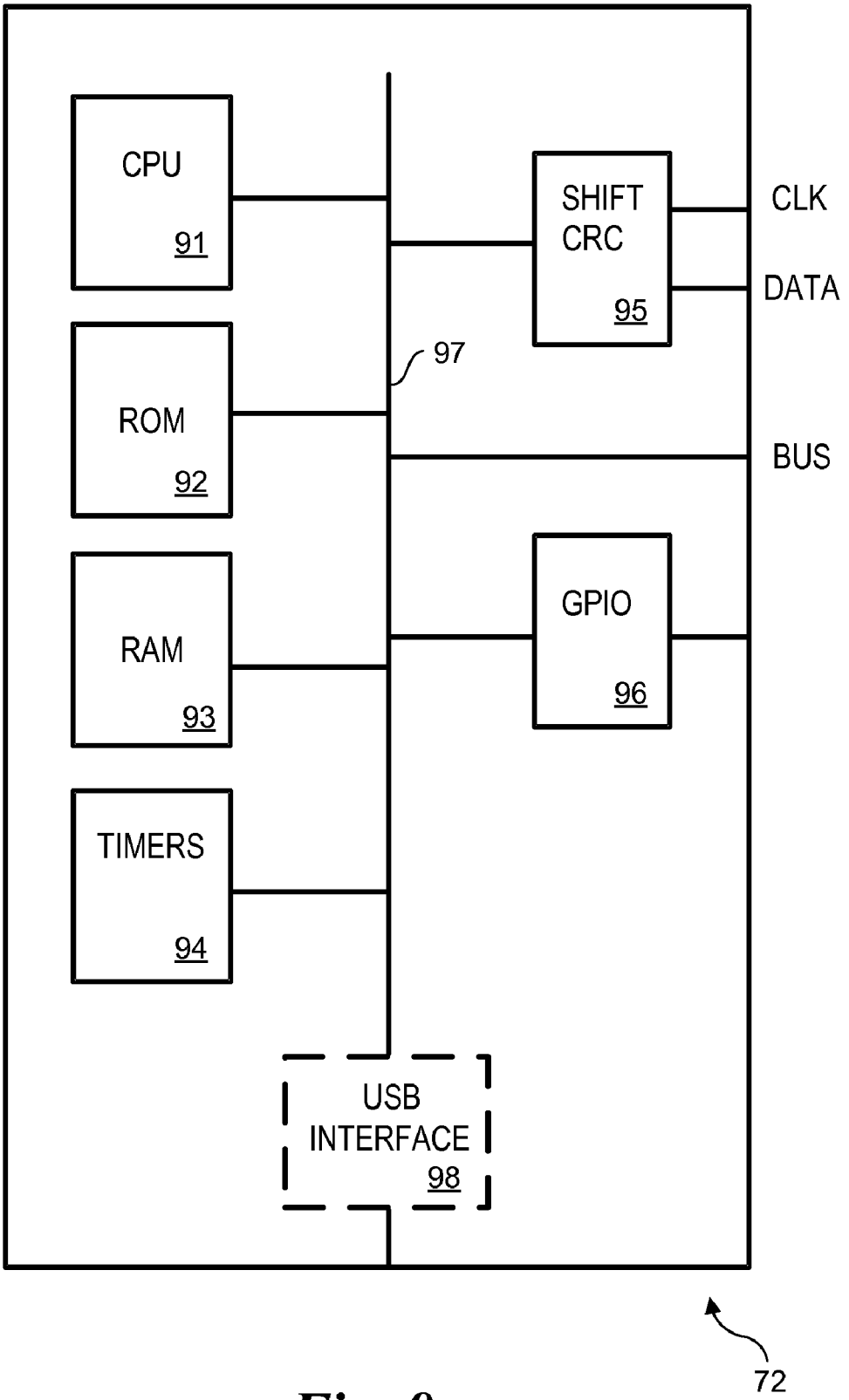

CPU
91
ROM
92
RAM
93
TIMERS
94
SHIFT
CRC
95
GPIO
96
USB
INTERFACE
98
97
CLK
DATA
BUS
72

Fig. 9

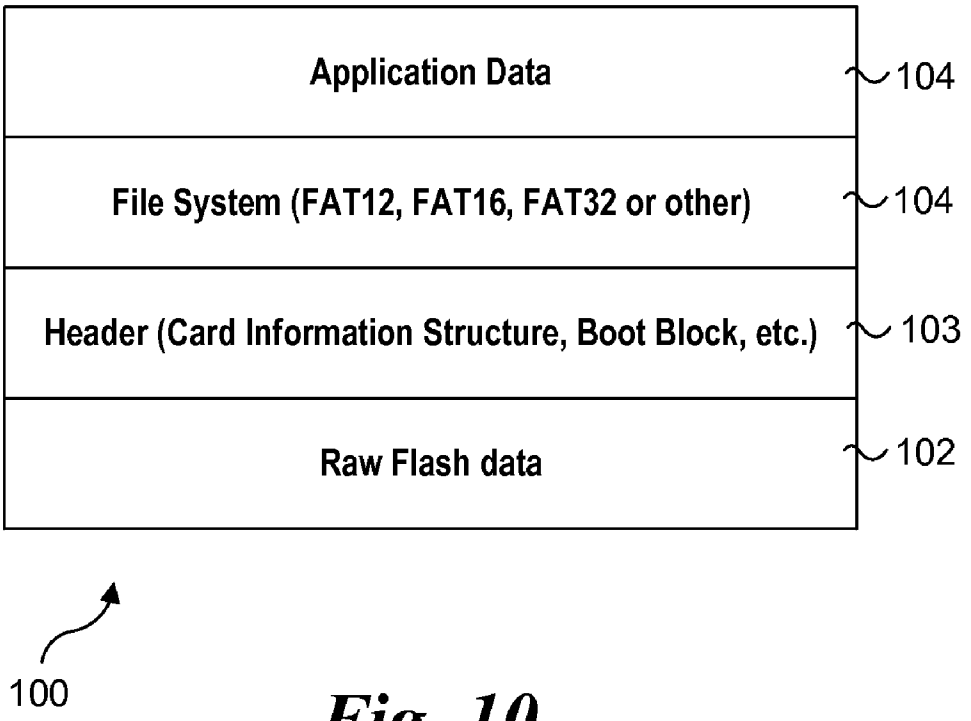

Application Data
104
File System (FAT12, FAT16, FAT32 or other)
104
Header (Card Information Structure, Boot Block, etc.)
103
Raw Flash data
102
100

Fig. 10

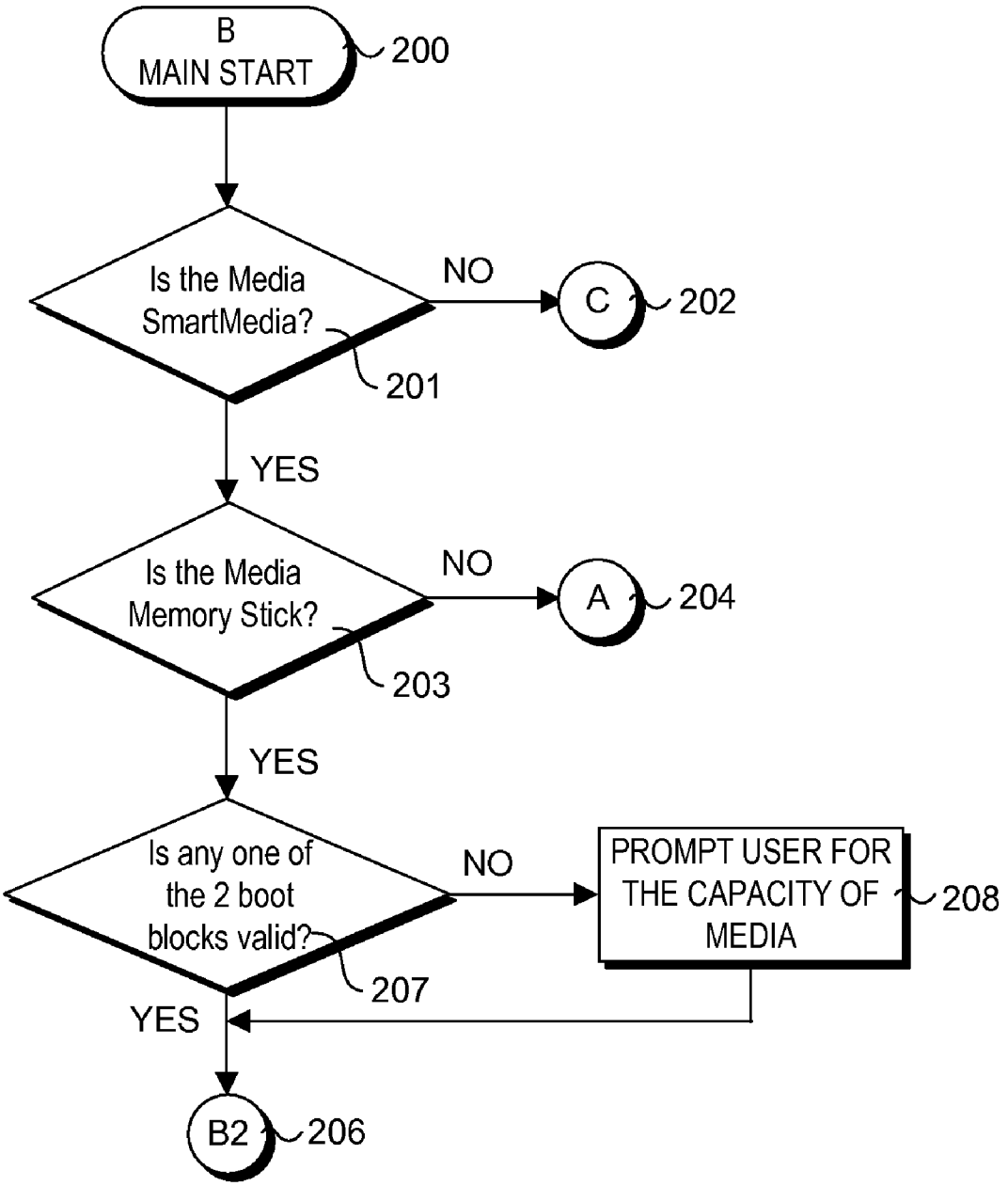

B
MAIN START
200
Is the Media SmartMedia?
NO
C
202
201
YES
Is the Media Memory Stick?
NO
A
204
203
YES
Is any one of the 2 boot blocks valid?
NO
PROMPT USER FOR THE CAPACITY OF MEDIA
208
207
YES
B2
206

Fig. 11

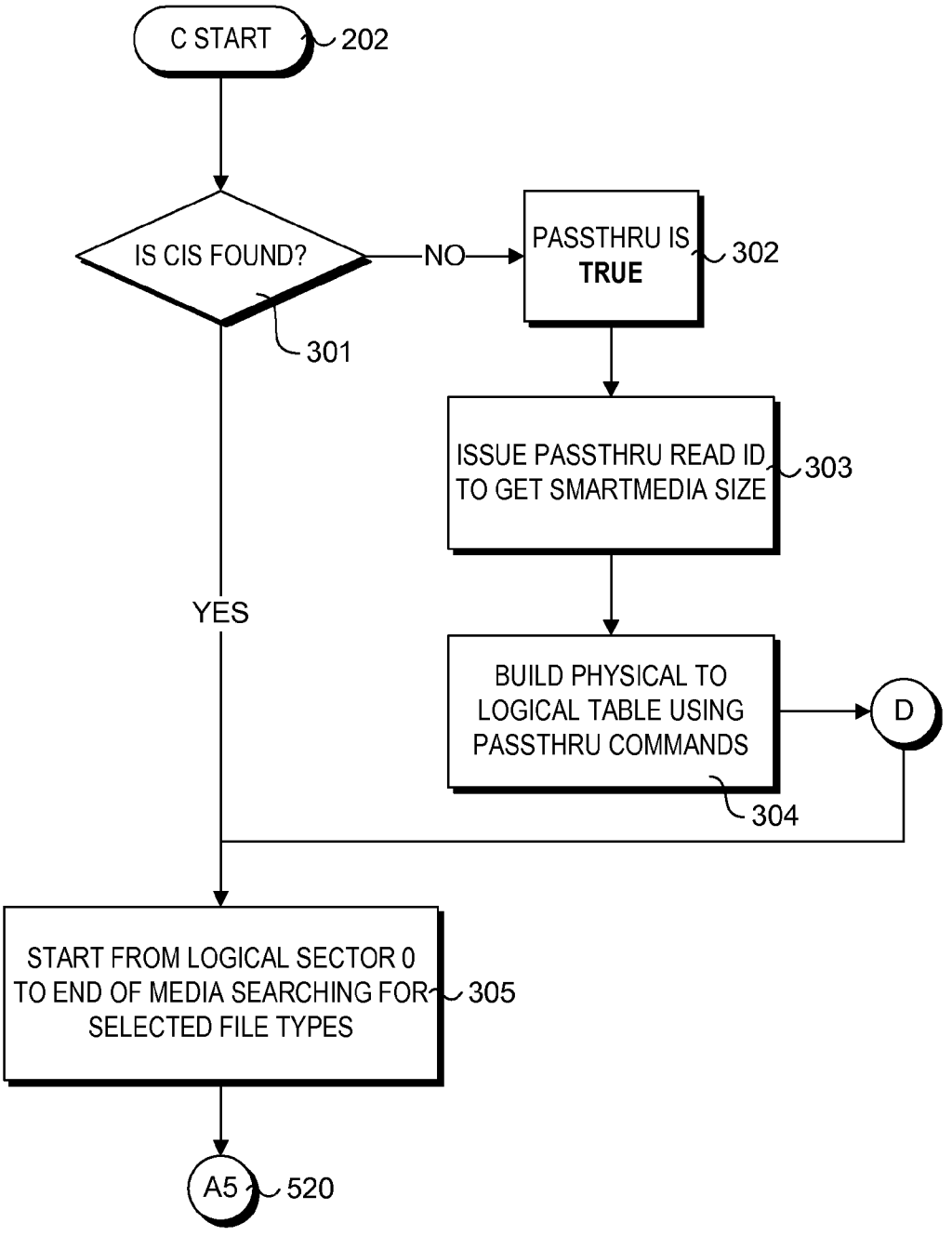

C START
202
IS CIS FOUND?
NO
PASSTHRU IS TRUE
302
301
ISSUE PASSTHRU READ ID TO GET SMARTMEDIA SIZE
303
YES
BUILD PHYSICAL TO LOGICAL TABLE USING PASSTHRU COMMANDS
D
304
START FROM LOGICAL SECTOR 0 TO END OF MEDIA SEARCHING FOR SELECTED FILE TYPES
305
A5
520

Fig. 12A

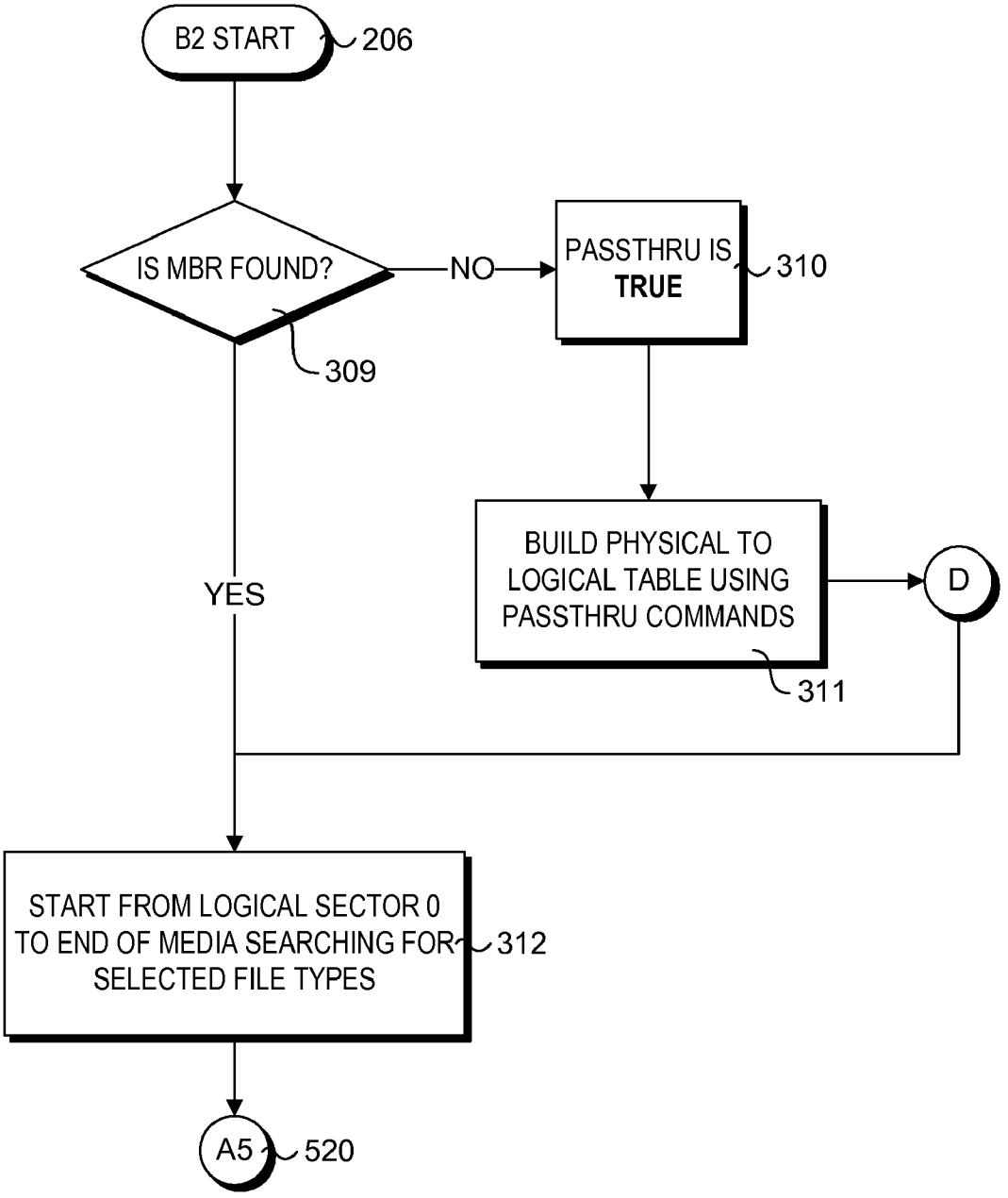

B2 START
206
IS MBR FOUND?
309
NO
PASSTHRU IS TRUE
310
YES
BUILD PHYSICAL TO LOGICAL TABLE USING PASSTHRU COMMANDS
311
D
START FROM LOGICAL SECTOR 0 TO END OF MEDIA SEARCHING FOR SELECTED FILE TYPES
312
A5
520

Fig. 12B

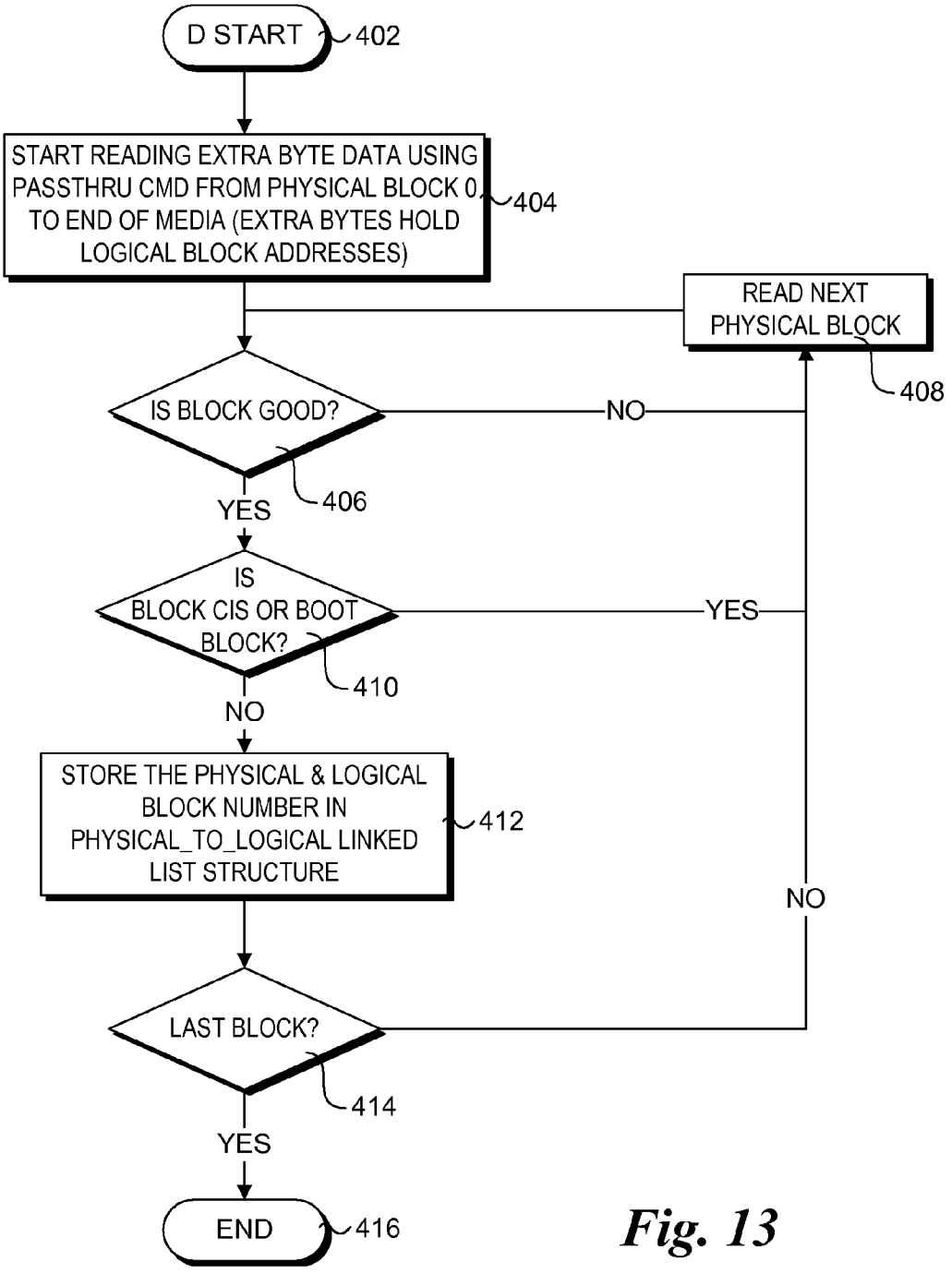

D START
402
START READING EXTRA BYTE DATA USING PASSTHRU CMD FROM PHYSICAL BLOCK 0 TO END OF MEDIA (EXTRA BYTES HOLD LOGICAL BLOCK ADDRESSES)
404
READ NEXT PHYSICAL BLOCK
408
IS BLOCK GOOD?
NO
406
YES
IS BLOCK CIS OR BOOT BLOCK?
YES
410
NO
STORE THE PHYSICAL & LOGICAL BLOCK NUMBER IN PHYSICAL_TO_LOGICAL LINKED LIST STRUCTURE
412
NO
LAST BLOCK?
414
YES
END
416

Fig. 13

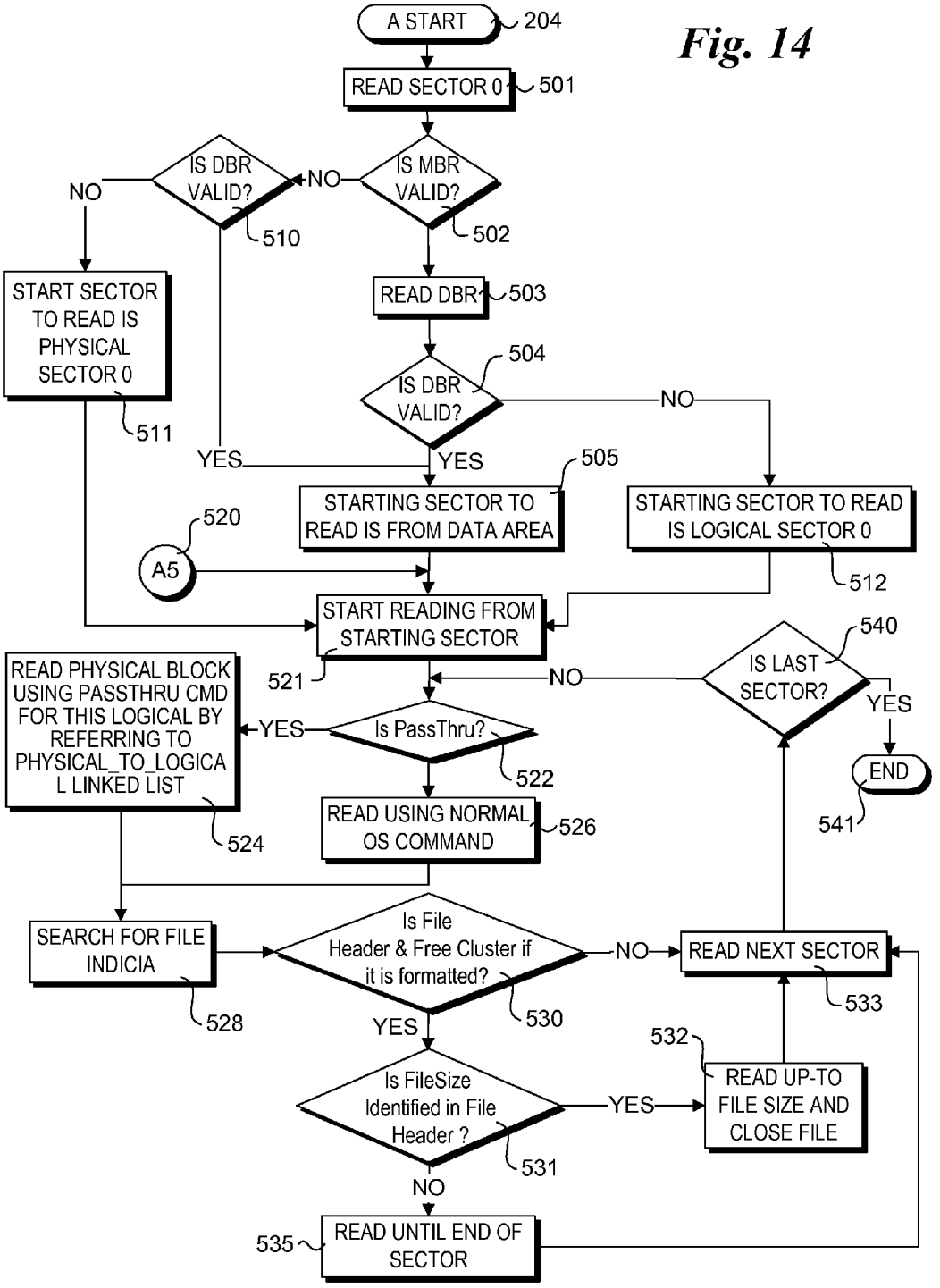

A START
204
READ SECTOR 0
501
IS MBR VALID?
502
NO
IS DBR VALID?
510
NO
START SECTOR TO READ IS PHYSICAL SECTOR 0
511
READ DBR
503
IS DBR VALID?
504
NO
YES
YES
STARTING SECTOR TO READ IS FROM DATA AREA
505
STARTING SECTOR TO READ IS LOGICAL SECTOR 0
512
A5
520
START READING FROM STARTING SECTOR
521
IS LAST SECTOR?
540
NO
YES
END
541
Is PassThru?
522
YES
READ PHYSICAL BLOCK USING PASSTHRU CMD FOR THIS LOGICAL BY REFERRING TO PHYSICAL_TO_LOGICAL LINKED LIST
524
READ USING NORMAL OS COMMAND
526
SEARCH FOR FILE INDICIA
528
Is File Header & Free Cluster if it is formatted?
530
NO
READ NEXT SECTOR
533
YES
Is FileSize Identified in File Header ?
531
YES
532
READ UP-TO FILE SIZE AND CLOSE FILE
NO
READ UNTIL END OF SECTOR
535

Fig. 14

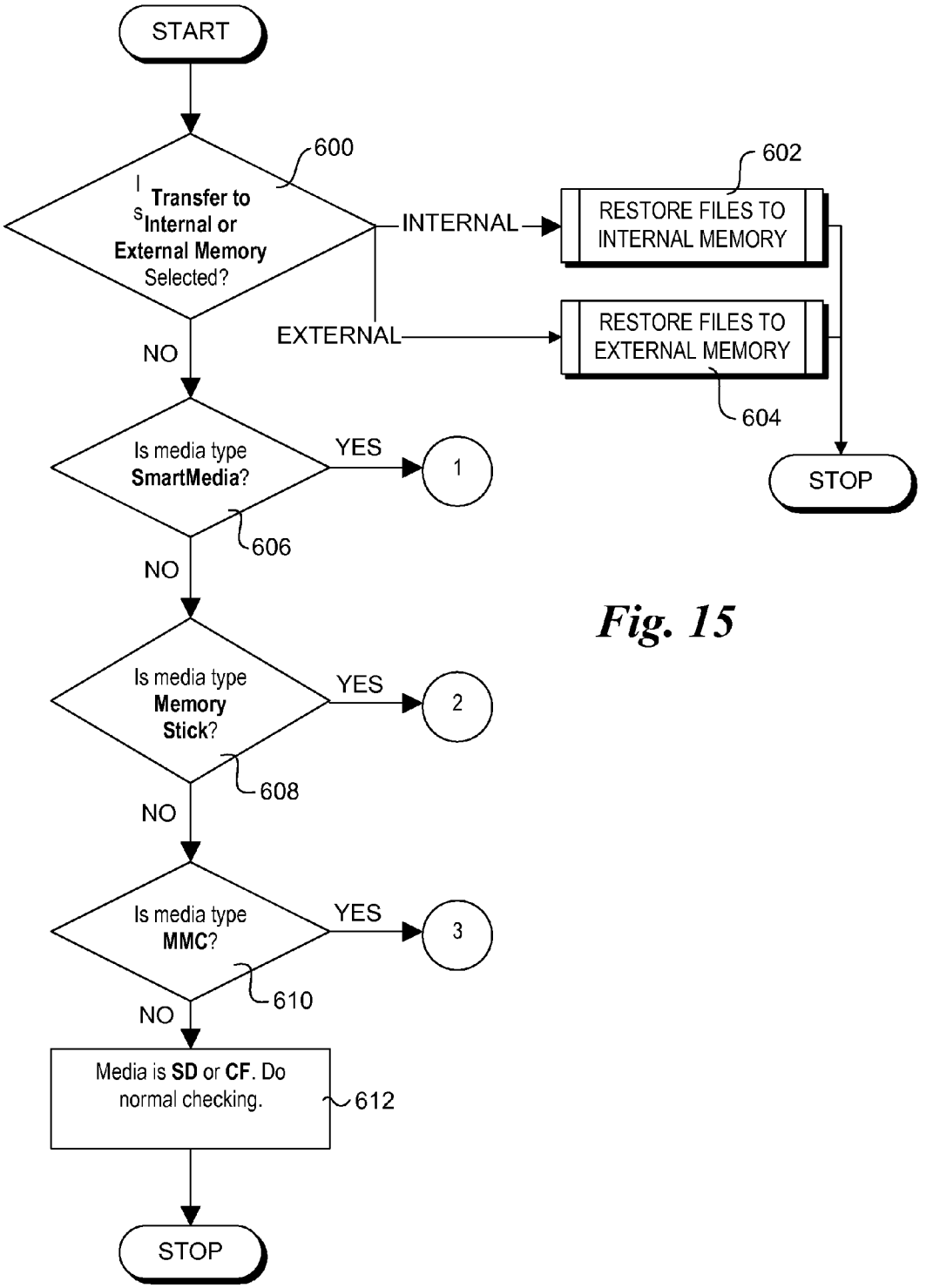

START
600
Is Transfer to Internal or External Memory Selected?
INTERNAL
602
RESTORE FILES TO INTERNAL MEMORY
EXTERNAL
RESTORE FILES TO EXTERNAL MEMORY
604
STOP
NO
Is media type SmartMedia?
YES
1
606
NO
Is media type Memory Stick?
YES
2
608
NO
Is media type MMC?
YES
3
610
NO
Media is SD or CF. Do normal checking.
612
STOP

Fig. 15

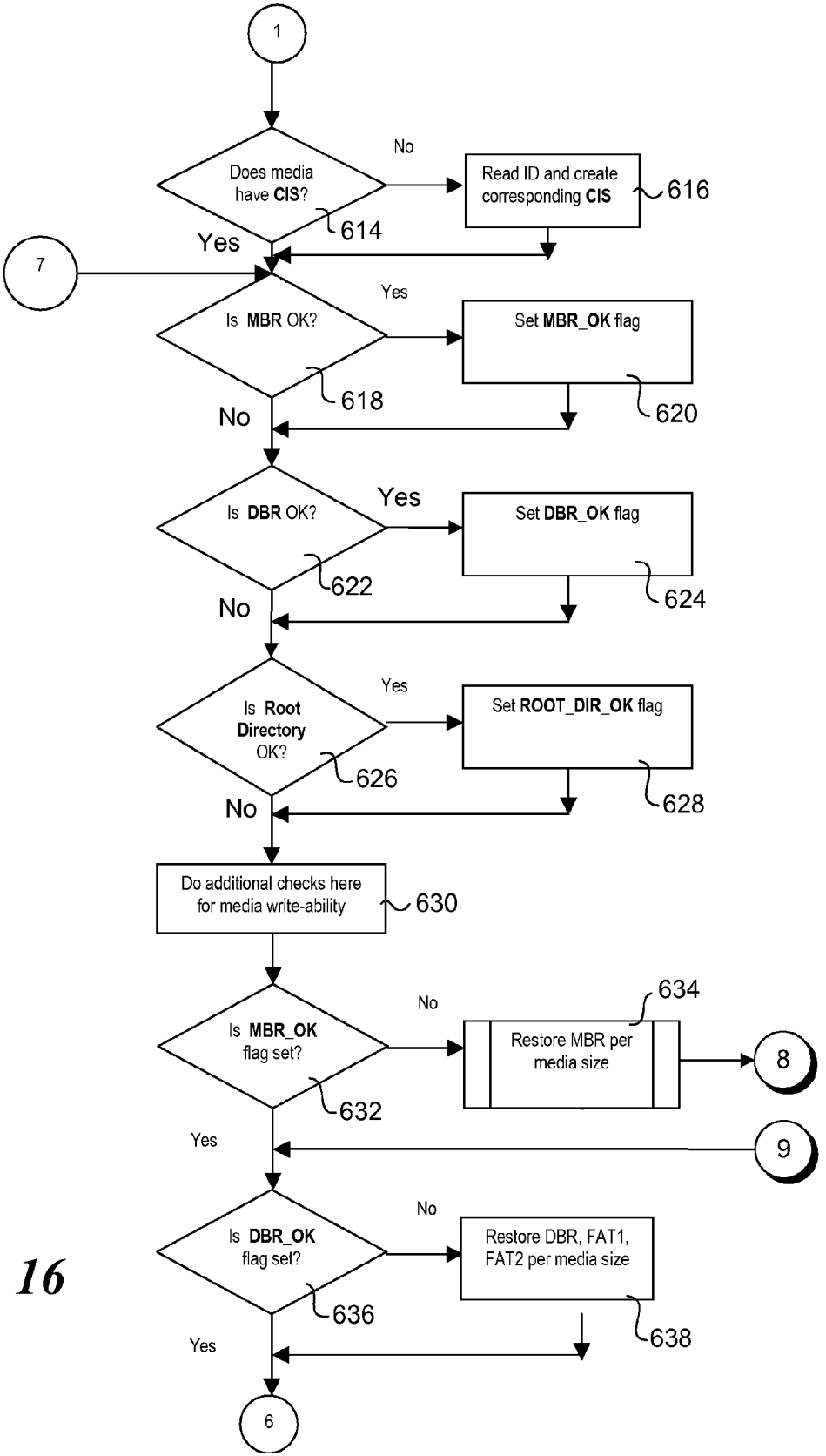

1
Does media have CIS?
614
No
Read ID and create corresponding CIS
616
Yes
7
Is MBR OK?
618
Yes
Set MBR_OK flag
620
No
Is DBR OK?
622
Yes
Set DBR_OK flag
624
No
Is Root Directory OK?
626
Yes
Set ROOT_DIR_OK flag
628
No
Do additional checks here for media write-ability
630
Is MBR_OK flag set?
632
No
634
Restore MBR per media size
8
Yes
9
Is DBR_OK flag set?
636
No
Restore DBR, FAT1, FAT2 per media size
638
Yes
6

Fig. 16

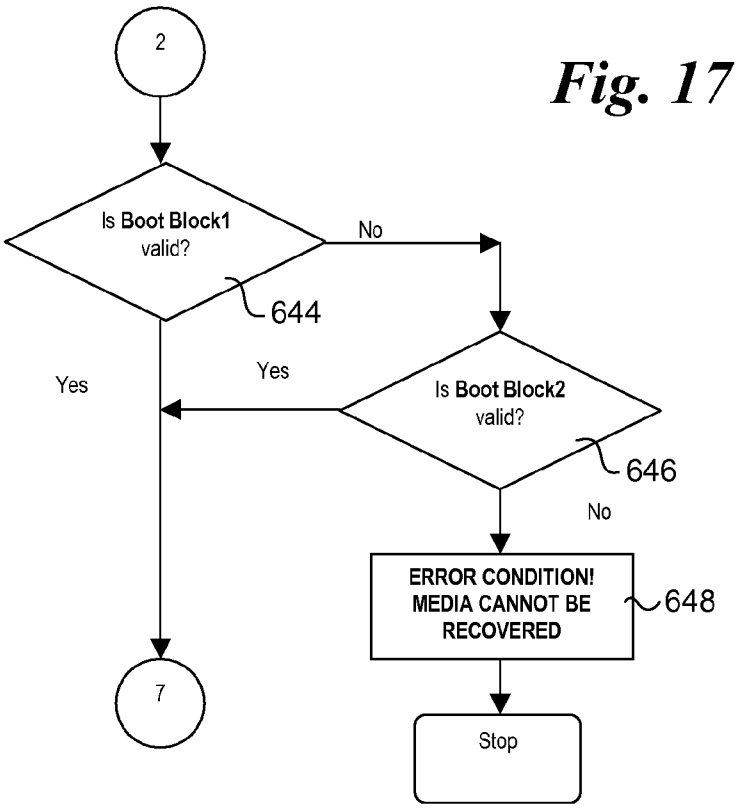

2
Is Boot Block1 valid?
644
No
Yes
Yes
Is Boot Block2 valid?
646
No
ERROR CONDITION! MEDIA CANNOT BE RECOVERED
648
7
Stop

Fig. 17

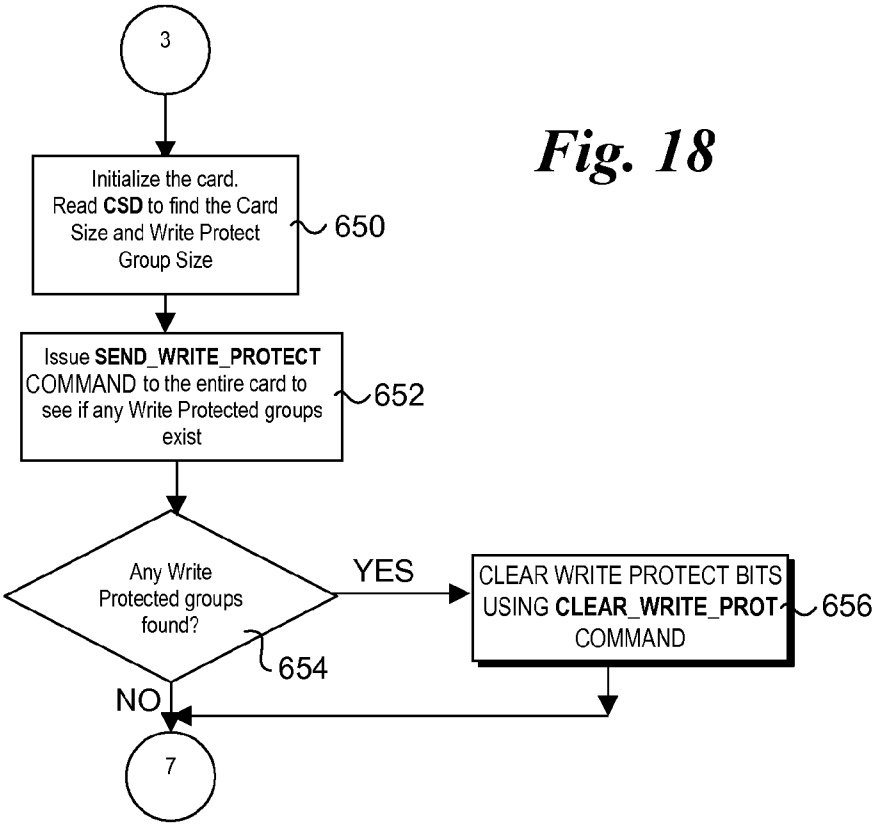

3
Initialize the card. Read CSD to find the Card Size and Write Protect Group Size
650
Issue SEND_WRITE_PROTECT COMMAND to the entire card to see if any Write Protected groups exist
652
Any Write Protected groups found?
654
YES
CLEAR WRITE PROTECT BITS USING CLEAR_WRITE_PROT COMMAND
656
NO
7

Fig. 18

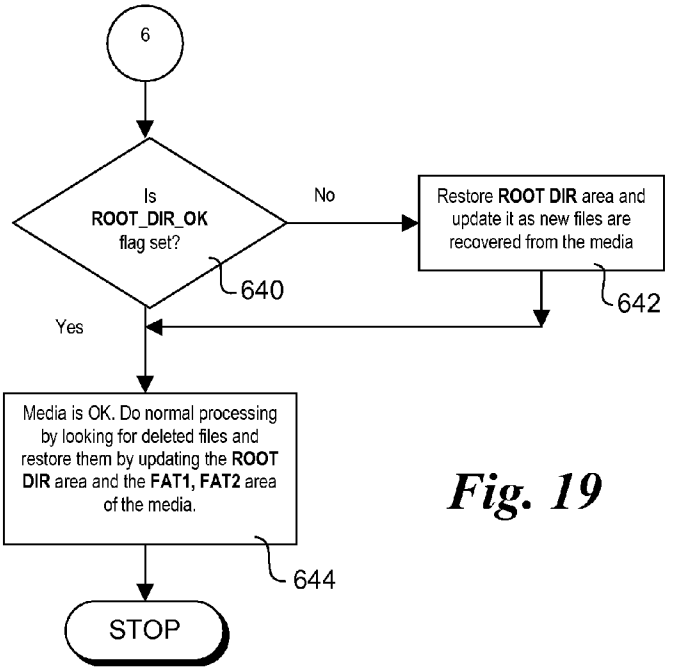

6
Is ROOT_DIR_OK flag set?
640
No
Restore ROOT DIR area and update it as new files are recovered from the media
642
Yes
Media is OK. Do normal processing by looking for deleted files and restore them by updating the ROOT DIR area and the FAT1, FAT2 area of the media.
644
STOP

Fig. 19

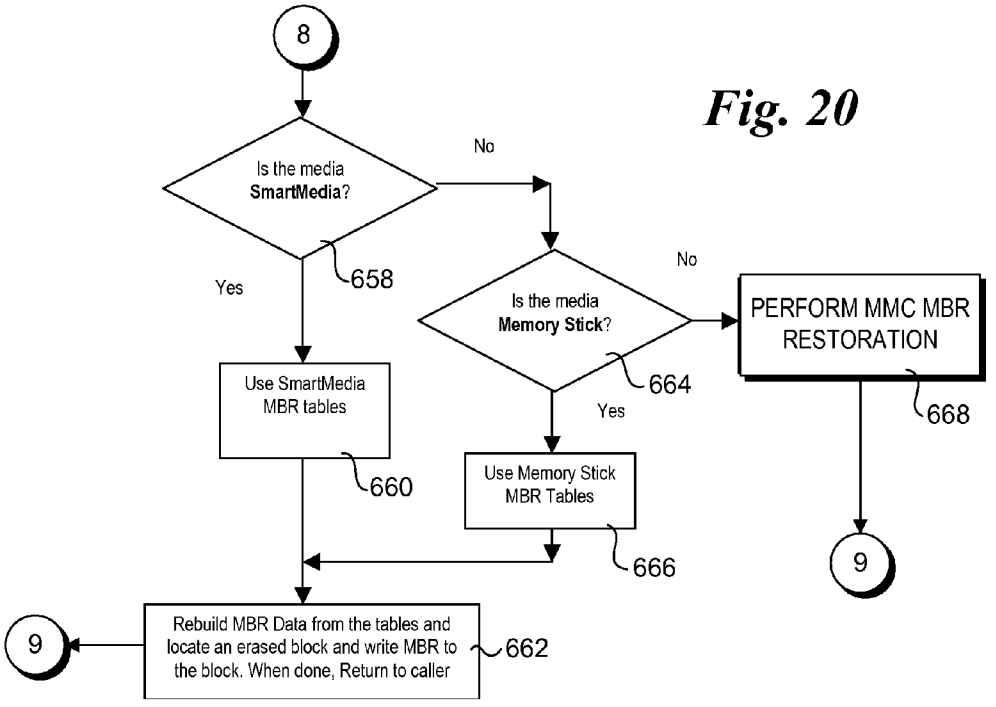

8
Is the media SmartMedia?
658
No
Yes
Is the media Memory Stick?
664
No
Yes
PERFORM MMC MBR RESTORATION
668
Use SmartMedia MBR tables
660
Use Memory Stick MBR Tables
666
9
9
Rebuild MBR Data from the tables and locate an erased block and write MBR to the block. When done, Return to caller
662

Fig. 20

Table 1. Block Details of SmartMedia

Size (MB)	MBR Page number	DBR Page Number	Number of Pages per Block	MBR as a Block Number	DBR as a Block Number	Root Directory Area as a Block Number	Start of Data Area as a Block Number
1	0	13	8	0	1	2	4
2	0	11	8	0	1	2	4
4	0	27	16	0	1	2	3
8	0	25	16	0	1	2	3
16	0	41	32	0	1	1	2
32	0	35	32	0	1	1	2
64	0	55	32	0	1	2	3
128	0	47	32	0	1	2,3	4
256	0	47	32	0	1-4	5	6

Fig. 21

Table 2. Block Details of Memory Stick

Size (MB)	MBR Page number	DBR Page Number	Number of Pages per Block	MBR as a Block Number	DBR as a Block Number	Root Directory Area as a Block Number	Start of Data Area as a Block Number
4	0	27	16	2	3	4,5	6
8	0	25	16	2	3	4,5	6
16	0	25	32	2	2	3	4
32	0	19	32	2	2	3	4
64	0	7	32	2	2	3	4
128	0	33	32	2	3,4	5	6

Fig. 22

FIELD-OPERABLE, STAND-ALONE APPARATUS FOR MEDIA RECOVERY AND REGENERATION

CROSS-REFERENCE TO RELATED APPLICATIONS

The present application is a continuation of U.S. Patent Application Ser. No. 11/075,496, filed Mar. 8, 2005, entitled "FIELD-OPERABLE, STAND-ALONE APPARATUS FOR MEDIA RECOVERY AND REGENERATION," which claims priority to PCT Publication WO 2004/027617 A1, published Apr. 1, 2004, entitled "FIELD-OPERABLE, STAND-ALONE APPARATUS FOR MEDIA RECOVERY AND REGENERATION," and which is a continuation-in-part of U.S. Pat. No. 6,839,864, issued Jan. 4, 2005, entitled "FIELD-OPERABLE, STAND-ALONE APPARATUS FOR MEDIA RECOVERY AND REGENERATION," which is a continuation-in-part of U.S. Pat. No. 7,093,161, issued Aug. 15, 2006, entitled "SOFTWARE RECOVERY METHOD FOR FLASH MEDIA WITH DEFECTIVE FORMATTING," and of U.S. Pat. No. 7,222,205, issued May 22, 2007, entitled "FLASHTOASTER FOR READING SEVERAL TYPES OF FLASH MEMORY CARDS WITH OR WITHOUT A PC," which is a continuation U.S. Pat. No. 6,438,638, issued Aug. 20, 2002, entitled "FLASHTOASTER FOR READING SEVERAL TYPES OF FLASH MEMORY CARDS WITH OR WITHOUT A PC," the specification and drawings of each of which are incorporated by reference herein in their entirety.

REFERENCE TO COMPUTER PROGRAM LISTING APPENDIX

A computer program listing appendix including details of various application program interface (API) commands employed by embodiments of the present invention was submitted with U.S. Patent Application Ser. No. 10/253,547 (now U.S. Pat. No. 6,839,864) on a compact disc. These materials are incorporated by reference herein in accordance with CFR 37 § 1.52(e)(5). The compact disc contained a single file entitled "API Listing.txt", created on Aug. 27, 2002, file size 37.0 KB.

FIELD OF THE INVENTION

The field of invention relates generally to flash media and, more specifically but not exclusively relates to a stand-alone apparatus for recovering data from defective flash media and regenerating the same.

BACKGROUND INFORMATION

Digital cameras typically capture images electronically and ultimately store the images as bits (ones and zeros) on a solid-state memory. Flash memory is the most common storage for digital cameras. Flash memory contains one or more electrically-erasable read-only-memory (EEPROM) integrated circuit chips that allow reading, writing, and block erasing.

Early digital cameras required the user to download or transfer the images from the flash memory within the digital camera to a personal computer (PC). A standard serial cable was most widely used. However, the limited transfer rate of the serial cable and the large size of the digital images made such serial downloads a patience-building experience. Serial downloads could easily take half an hour for only a few dozen images.

Digital camera manufacturers solved this problem by placing the flash memory chips on a small removable card. The flash-memory card could then be removed from the digital camera, much as film is removed from a standard camera. The flash-memory card could then be inserted into an appropriate slot in a PC, and the image files directly copied to the PC.

FIG. **1**A shows a conventional scheme in which a flash memory card and adapter are employed for transferring images from a digital camera to a PC. Generally, digital camera **14** will employ a flash memory card containing one or more flash memory chips that are used to store data pertaining to images taken by users of the camera. In the illustrated example, the flash memory chip(s) are contained in a CompactFlash™ card **16** (CompactFlash is a trademark of SanDisk Corp. of Sunnyvale, Calif.), which can be removed from digital camera **14** by pressing a card-eject button.

Typically, laptop and notebook computers have one or more PC-card (earlier known as PCMCIA, Personal Computer Memory Card International Association) slots that can receive PCMCIA cards. In this conventional scheme, a CF-to-PCMCIA adapter **10** is employed to enable a PC or laptop computer to read data stored on CompactFlash card **16** as if it were stored on a PCMCIA card. As shown in FIG. **1**A, CF-to-PCMCIA adapter **10** contains a connector interface disposed in an opening that receives CompactFlash card **16**. FIG. **1**B shows CF-to-PCMCIA adapter **10** with CompactFlash card **16** inserted.

As shown in FIG. **1**C, a computer such as notebook PC **20** can read data stored on a PCMCIA card via either a PCMCIA reader **12** connected to the PC via a parallel or USB cable, or directly via a built-in PCMCIA slot **22**. Once the PCMCIA is inserted into the reader or the PCMCIA slot, the user merely has to copy the image files from CompactFlash card **16** to the hard disk of PC **20**. Since high-speed parallel buses are used, transfer is rapid, about the same speed as accessing the hard disk. This greatly enhances the transfer speed when compared with the serial cable link discussed above.

Although the CompactFlash card format is relatively small, being not much more than an inch square, other smaller cards have recently emerged. For example, several of such smaller flash media cards are shown in FIG. **2**A, including a SmartMedia™ (SmartMedia is a trademark of the SSFDC Forum of Tokyo, Japan) card **24**, which is less than half an inch long, yet has enough flash memory capacity for dozens of images. The SmartMedia card may be read in a manner similar to that discussed above with a SmartMedia-to-PCMCIA adapter **10'**. Depending on the manufacturer and card capacity, different adapters **10'** may be required for different memory capacities of SmartMedia card **24**.

Other kinds of flash-memory cards that are being championed by different manufacturers include MultiMediaCard™ (MMC) **28** and the related Secure Digital Card (SD) **26**. MMC is a trademark of SanDisk Corp. of Sunnyvale, Calif. and Infineon Systems, Germany, while SD is controlled by the SD Group that includes Matsushita Electric Industrial Co., SanDisk Corporation, Toshiba Corp. Another emerging form factor from Sony is Memory Stick **18**. Typically, memory stick devices may be read using a PCMCIA/Floppy adapter while MMC may be read with a floppy adapter.

The different physical shapes and pin arrangements of cards **24**, **26**, **28** and Memory Stick **18** prevent their use in CF-to-PCMCIA adapter **10**. Indeed, most of these cards **24**, **26**, **28** have less than a dozen pins, while CompactFlash card **16** has a larger 50-pin interface. Furthermore, serial data interfaces are used in the smaller cards **24**, **26**, **28**, while a parallel data bus is used with CompactFlash card **16**.

FIG. **2**B shows a Memory Stick-to-PCMCIA adapter that includes an active converter chip **11**. Memory Stick **18** fits into an opening in Memory Stick-to-PCMCIA adapter **15**, allowing adapter **15** and the Memory Stick to be plugged into a standard PCMCIA slot on a PC in a manner similar to CF-to-PCMCIA adapter **10**. However, rather than providing a passive adaptor function, adapter **15** employs active converter chip **11** to convert the serial data format of Memory Stick **18** to the parallel data format of a 68-pin PCMCIA slot. Inclusion of converter chip **11** in adapter **15** significantly increases the cost and complexity of adapter **15** compared to CF-to-PCMCIA adapter **10**.

While the advances in flash-memory card technology are useful, the many different card formats present a confusing array of interface requirements to a PC. Different adapters are needed for each of the card formats. PCMCIA card reader **12** can be replaced with other format readers, such as a SmartMedia Card reader, and even some multi-standard readers are available, such as a universal reader from Lexar Media that reads CompactFlash or SmartMedia in addition to PCMCIA.

Occasionally, a user may remove a flash media from a host device, such as a digital camera, PDA (Personal Digital Assistant), MP3 player, etc., while the host device is reading to or writing from the card. Media card removal, particularly during the writing, but also sometimes during the reading, may destroy the card formatting, resulting, in most cases, in making the data unreadable by the host and possibly other interfacing devices. In addition, flash media card formatting and file information may become corrupted or lost due to other causes, such as power failure during read or write operations, faulty programs, faulty host device or reader, high level "erasing" of files by a user, who later wants to recover such files, or other causes. As a result, the content on the card cannot be read and/or accessed.

Currently, there exists techniques for recovering data from such media through use of a computer and (generally) some time of flash media "reader" or other type of flash media interface device. However, when someone is using a device that stores data on flash media in the field, such as a tourist using a digital camera on an overseas trip, the user may not have access to a computer and/or reader. What is clearly needed is a field-operable, stand-alone apparatus that allows the regeneration and recovery of corrupted media of all types, without having to rely on the availability of a computer or reader. Furthermore, it would be advantageous if such an apparatus could read a variety of different card formats.

SUMMARY OF THE INVENTION

In accordance with aspects of the present invention a method for recovering data from corrupted flash media and regenerated damaged flash media is disclosed. In one embodiment, the method comprises accessing data on a flash media, searching the data for file indicia corresponding at least one predetermined file type, reading the data based on information in the file indicia, the data including a recovered file, and storing the recovered file on a storage medium.

BRIEF DESCRIPTION OF THE DRAWINGS

The foregoing aspects and many of the attendant advantages of this invention will become more readily appreciated as the same becomes better understood by reference to the following detailed description, when taken in conjunction with the accompanying drawings, wherein like reference numerals refer to like parts throughout the various views unless otherwise specified:

FIG. **1**A shows a flash memory card and adapter for transferring images from a digital camera to a PC corresponding to a conventional transfer scheme;

FIG. **1**B shows the CF-to-PCMCIA adapter with a CompactFlash card fully inserted;

FIG. **1**C shows a PC connected to a PCMCIA reader;

FIG. **2**A illustrates various formats of flash-memory cards used with digital cameras;

FIG. **2**B shows a Memory Stick-to-PCMCIA adapter that employs an active converter chip;

FIG. **3**A shows a universal CompactFlash adapter that accepts SmartMedia, MultiMediaCard, Secure Digital, and Memory Stick flash-memory cards;

FIG. **3**B shows a CompactFlash reader that reads SmartMedia, MultiMediaCard, Secure Digital, and Memory Stick flash-memory cards through passive adapters to the CompactFlash form factor;

FIG. **3**C shows a universal CompactFlash adapter that accepts SmartMedia-compatible cards, MultiMediaCard-compatible cards, Secure Digital-compatible cards, Memory Stick-compatible cards, and USB Drive;

FIG. **3**D shows a CompactFlash reader that reads SmartMedia-compatible cards, MultiMediaCard-compatible cards, Secure Digital-compatible cards, and Memory Stick-compatible cards through passive adapters to the CompactFlash form factor;

FIGS. **4**A-E show card-type detection using the A**1**, A**0** pins of the CompactFlash reader interface;

FIG. **5** is a table of pin mappings for the SmartMedia, MMC/SD, and Memory Stick to CompactFlash adapters;

FIG. **6** is a diagram of a multi-slot embodiment of the flash-card reader;

FIG. **7** is a schematic diagram of a field operable stand-alone apparatus in accordance with an embodiment of the invention that may be used to recover data from damaged flash media and/or regenerate such media so that it may be used again by a host device;

FIG. **8** is a schematic diagram of an optional configuration of the field operable apparatus of FIG. **7**;

FIG. **9** is a schematic block diagram illustrating the major functional components of an exemplary controller that may be employed in the field operable apparatus of FIGS. **7** and **8**;

FIG. **10** is a block diagram illustrating a data stack corresponding to a typical flash media data access scheme;

FIG. **11** is a flowchart showing high-level operations corresponding to a software program in accordance with an embodiment of the invention;

FIG. **12***a* is a continuation of the flowchart of FIG. **11** illustrating operations and logic pertaining to processing a SmartMedia flash device;

FIG. **12***b* is a continuation of the flowchart of FIG. **11** illustrating operations and logic pertaining to processing various types of flash media that don't include SmartMedia and Memory Stick devices;

FIG. **13** is a continuation of the flowchart of FIG. **11** illustrating operations and logic performed when building a Physical-to-Logical table that maps physical storage locations to logical storage locations;

FIG. **14** is a continuation of the flowchart of FIG. **11** illustrating operations and logic performed when reading data corresponding to previously inaccessible files, such as picture files and the like;

FIGS. **15-20** collectively comprise a flowchart illustrating operations and logic performed during media regeneration in accordance with one embodiment of the invention; and

FIGS. **21** and **22** respectively show tables containing parameters specific to SmartMedia and Memory Stick type media that are used during media regeneration operations.

DETAILED DESCRIPTION OF PREFERRED EMBODIMENTS

Embodiments of apparatus for recovering data from and regenerating corrupted flash media are described herein. In the following description, numerous specific details are set forth to provide a thorough understanding of embodiments of the invention. One skilled in the relevant art will recognize, however, that the invention can be practiced without one or more of the specific details, or with other methods, components, materials, etc. In other instances, well-known structures, materials, or operations are not shown or described in detail to avoid obscuring aspects of the invention.

Reference throughout this specification to "one embodiment" or "an embodiment" means that a particular feature, structure, or characteristic described in connection with the embodiment is included in at least one embodiment of the present invention. Thus, the appearances of the phrases "in one embodiment" or "in an embodiment" in various places throughout this specification are not necessarily all referring to the same embodiment. Furthermore, the particular features, structures, or characteristics may be combined in any suitable manner in one or more embodiments.

In accordance with aspects of the invention, details for making and using a field-operable stand-alone apparatus for recovering data from damaged flash media and regenerated such damaged flash media are disclosed herein. Included among these aspects, is the ability of embodiments of the apparatus to enable recovery from and regeneration of various types of flash media. In one embodiment, this is enabled via use of a universal flash media adapter.

For example, FIG. **3**A shows a universal CompactFlash adapter that accepts SmartMedia, MultiMediaCard, Secure Digital, and Memory Stick flash-memory cards. Digital camera **14** stores images on flash memory that is in one of several card types. CompactFlash card **16** uses a 50-pin connector and transfers image data in a 16-bit parallel format.

SmartMedia card **24** comprises a smaller flash-memory card with a 22-pin interface and transfers data in an 8-bit parallel format. SmartMedia adapter **30** converts the 22-pin SmartMedia interface to fit within the 50-pin CompactFlash interface. When SmartMedia card **24** is plugged into SmartMedia adapter **30**, both can be plugged into a CompactFlash slot on a CompactFlash reader. Of course, ordinary CompactFlash readers will not be able to read SmartMedia card **24** since special signal conversion is required by the CompactFlash reader. Furthermore, the SmartMedia Card **24** includes the Extreme Digital card (xD Card), or any other flash memory card built from solid state memory that is electrically similar to the SmartMedia **24**.

MultiMediaCard **28** and Secure Digital card **26** are flash-memory cards with similar 9-pin interfaces. Serial data transfer is used through a single Data I/O pin. MMC/SD adapter **32** has an opening with a 9-pin connector to receive either MultiMediaCard **28** or Secure Digital card **26**. Once MultiMediaCard **28** or Secure Digital card **26** is inserted into MMC/SD adapter **32**, the MMC/SD adapter **32** can be inserted into a CompactFlash slot on a special CompactFlash reader. The CompactFlash reader then detects the card type and performs serial-to-parallel conversion. In some embodiments, the MultiMediaCard **28** is accessible in serial (1-bit mode) or in parallel (more than 1-bit modes). Furthermore, the MultiMediaCard **28** includes at least one of Reduced-size MultiMediaCard (RS MMC), MultiMediaCard High Speed (MMC HS), or any other flash memory card electrically similar to the MultiMediaCard **28**. In some embodiments, the Secure Digital Card **26** is accessible in serial (1-bit mode) or in parallel (more than 1-bit modes). Furthermore, the Secure Digital Card **26** includes at least one of mini Secure Digital Card (miniSD), Secure Digital Card High Speed (SD HS), or any other flash memory card electrically similar to the Secure Digital Card **26**.

Memory Stick **18** is also a flash-memory card with a 10-pin, serial-data interface, but is narrower and longer than MultiMediaCard **28** or Secure Digital card **26**. Memory Stick adapter **34** has an opening with a 10-pin connector to receive Memory Stick **18**. Once Memory Stick **18** is inserted, Memory Stick adapter **32** can itself be inserted into a CompactFlash slot on a special CompactFlash reader. The CompactFlash reader then detects the card type and performs serial-to-parallel conversion.

FIG. **3**B shows a CompactFlash reader that reads SmartMedia, MultiMediaCard, Secure Digital, and Memory Stick flash-memory cards through passive adapters to the CompactFlash form factor. CompactFlash reader **42** has an opening or slot with 50-pin connector **44** that accepts CompactFlash card **16**. Controller chip **40** performs handshaking with CompactFlash card **16** and performs data transfer. CompactFlash reader **42** also connects to a PC over USB connector **46**. Controller chip **40** also controls the USB interface to the host PC, allowing image files to be transferred to the PC from CompactFlash card **16**.

Other kinds of flash-memory cards can also be read by CompactFlash reader **42**. For example, adapter **34** allows Memory Stick **18** to be read. Memory Stick adapter **34** has an opening that Memory Stick **18** fits into, while Memory Stick adapter **34** itself fits into 50-pin connector **44**, since adapter **34** has the same form factor as a CompactFlash card. SmartMedia card **24** can also be read by CompactFlash reader **42**, using SmartMedia adapter **30**. Likewise, MultiMediaCard **28** or Secure Digital card **28** can be read using MMC/SD adapter **32**.

Adapters **30**, **32**, **34** are passive adapters that only connect pins from the smaller flash-memory cards to the 50-pin CompactFlash connector. An active converter chip is not required, greatly reducing cost and complexity.

FIG. **3**C shows an alternate embodiment of a universal CompactFlash memory adapter that accepts SmartMedia-compatible flash memory cards, MultiMediaCard-compatible flash memory cards, Secure Digital-compatible flash memory cards, Memory Stick-compatible flash-memory cards, and Universal Serial Bus (USB) Drive. Digital camera **3014** stores images on flash memory that is in one of several card types. CompactFlash card **3016** uses a 50-pin connector and transfers image data in a 16-bit parallel format.

Some examples of SmartMedia-compatible flash memory cards **3024** include NAND Multi-Level Cell (MLC) flash memory, Assist-Gate-AND (AG-AND) flash memory, xD Card, NAND Single-Level Cell (SLC) flash memory, etc. In some embodiments, a SmartMedia adapter **3030** converts the 22-pin SmartMedia interface to fit within the 50-pin CompactFlash interface. When SmartMedia-compatible card **3024** is plugged into SmartMedia adapter **3030**, both can be plugged into a CompactFlash slot on a CompactFlash reader. Of course, ordinary CompactFlash readers will not be able to read SmartMedia-compatible card **3024** since special signal conversion is required by the CompactFlash reader. Details of the special signal conversion are discussed below.

MultiMediaCard-compatible card **3028** and Secure Digital-compatible card **3026** are flash-memory cards with similar

9-pin interfaces. Some examples of MultiMediaCard-compatible cards include Reduced size MultiMediaCard (RS MMC), MultiMediaCard High Speed (MMC HS), etc. Some examples of Secure Digital-compatible cards include mini-Secure Digital cards (miniSD), Secure Digital High Speed (SD HS), TransFlash, T-Flash, etc. Serial data transfer is used through a single Data I/O pin. MMC/SD adapter **3032** has an opening with a 9-pin connector to receive either a MultiMediaCard-compatible card **3028** or a Secure Digital-compatible card **3026**. Once MultiMediaCard-compatible **3028** or Secure Digital-compatible card **3026** is inserted into MMC/SD adapter **3032**, the MMC/SD adapter **3032** can be inserted into a CompactFlash slot on a special CompactFlash reader. The CompactFlash reader then detects the card type and performs serial-to-parallel conversion. Alternatively, the MMC-compatible card is accessible in serial (1-bit mode) or in parallel (more than 1-bit modes). Likewise, the Secure Digital-compatible cards may be accessed in serial or in parallel.

Memory Stick-compatible card **3018** is also a flash-memory card with a 10-pin, serial-data interface, but may be narrower and longer than MultiMediaCard-compatible **3028** or Secure Digital-compatible card **3026**. One example of a Memory Stick-compatible card **3018** is Memory Stick Duo (MS Duo). In one embodiment, Memory Stick adapter **3034** has an opening with a 10-pin connector to receive Memory Stick-compatible card **3018**. Once Memory Stick-compatible card **3018** is inserted, Memory Stick adapter **3032** can itself be inserted into a CompactFlash slot on a special CompactFlash reader. The CompactFlash reader then detects the card type and performs serial-to-parallel conversion.

Alternatively, a USB Drive **3025** having a similar form factor to the flash memory cards described above can be used. The USB Drive **3025** may include a storage device built from at least one of a magnetic medium, a NAND SLC flash, an AG-AND flash, a NAND MLC flash, or any flash media device having a similar electrical interface. Furthermore, the USB Drive **3025** may include one or more chips built from at least one of NAND SLC flash, AG-AND flash, NAND MLC flash, or any flash media device having a similar electrical interface. The one or more chips may be cascaded to optimize for one or more operations, such as Read operation, Write operation, etc.

FIG. **3**D shows a CompactFlash reader that reads SmartMedia-compatible cards, MultiMediaCard-compatible cards, Secure Digital-compatible cards, and Memory Stick-compatible flash-memory cards through passive adapters to the CompactFlash form factor. CompactFlash reader **3042** has an opening or slot with 50-pin connector **3044** that accepts CompactFlash card **3016**. Controller chip **40** performs handshaking with CompactFlash card **3016** and performs data transfer. CompactFlash reader **3042** also connects to a PC over a USB connector **46**. Controller chip **40** may control the USB interface to the host PC, allowing image files to be transferred to the PC from CompactFlash card **3016**.

Other kinds of flash-memory cards can also be read by CompactFlash reader **3042**. For example, adapter **3034** allows Memory Stick-compatible card **3018** to be read. Memory Stick adapter **3034** has an opening that Memory Stick-compatible card **3018** fits into, while Memory Stick adapter **3034** itself fits into the 50-pin connector **3044**, since adapter **3034** has the same form factor as a CompactFlash card. SmartMedia-compatible card **3024** can be read by CompactFlash reader **3042**, using SmartMedia adapter **3030**. Likewise, MultiMediaCard-compatible card **3028** or Secure Digital-compatible card **3028** can be read using MMC/SD adapter **3032**.

Adapters **3030**, **3032**, **3034** are passive adapters that only connect pins from the smaller flash-memory cards to the 50-pin CompactFlash connector. An active converter chip is not required, greatly reducing cost and complexity.

FIGS. **4**A-E detail detection of the type of flash-memory card by the CompactFlash reader. Since the same CompactFlash slot is used for many kinds of flash-memory cards, a detection method is useful so that the user doesn't have to explicitly indicate what type of flash-memory card is inserted into the CompactFlash reader.

The inventors have carefully examined the pins of the interfaces to the various flash-memory cards and have discovered that type-detection can be performed by examining two address pins. Address pins A**0** and A**1** are the least-significant-bits (LSB) of the address of the 50-pin CompactFlash interface. These pins are normally inputs to the CompactFlash card and thus are driven by the CompactFlash reader. When the reader does not drive A**0**, A**1** to the inserted CompactFlash card, the A**0**, A**1** pins float or are pulled high by pull-up resistors.

Address pins are not present on the other kinds of flash-memory cards. Instead, the address and data are multiplexed. For MMC/SD and Memory Stick, the address is sent serially. Using the adapters, pins from the other flash-memory cards can be connected to the CompactFlash pins. Pins A**0** and A**1** are used to detect the type of card. For SmartMedia, the addresses are sent by using a special control sequence followed by 3 or 4 bytes of starting address.

In FIG. **4**A, the A**1**, A**0** pins of the CompactFlash reader interface are highlighted. Converter chip **40** in the CompactFlash reader normally drives all 11 address pins in the CompactFlash interface when reading a CompactFlash card plugged into connector **44**. The A**0** pin from the CompactFlash card plugs into connector cup **56**, while the A**1** pin from the CompactFlash card plugs into connector cup **58** of 50-pin connector **44**.

Card-type detector **50** has two pull-up resistors added to lines A**0**, A**1**. Resistor **52** pulls line A**0** high to power (Vcc) when neither converter chip **40** nor a card plugged into connector **44** drives line A**0**. Likewise, resistor **54** pulls line A**1** high when line A**1** is not being actively driven. During detection mode, converter chip **40** is programmed to not drive lines A**0**, A**1** and instead use then as inputs to the detector logic.

In FIG. **4**B, a CompactFlash card is inserted into the connector for card-type detection. CompactFlash card **16** is plugged into connector **44**. Since A**0** and A**1** are inputs to CompactFlash card **16**, they are not driven by CompactFlash card **16**. During detection mode, converter chip **40** also does not drive pins A**0**, A**1**. Thus lines A**0**, A**1** are left floating and are each pulled high by resistors **52**, **54**.

Detection logic in converter chip **40** reads card-select pins CD**0**, CD**1** to detect the presence of a flash-memory card. When a new card is present, detection logic then reads pins A**0**, A**1** as inputs. Both inputs are high. The detection logic in converter chip **40** recognizes the HH state of A**0**, A**1** as indicating that a CompactFlash card is plugged into connector **44**. Converter chip **40** then exits detection mode and configures its interface to connector **44** for the 50-pin CompactFlash interface as shown later in FIG. **5**.

In FIG. **4**C, a MultiMediaCard or Secure Digital card is inserted into the connector for card-type detection. MMC/SD card **28** (not shown) is plugged into MMC/SD adapter **32** which is plugged into connector **44**. Converter chip **40** does not drive pins A**1**, A**0** during detection mode. Thus pin A**1** floats and is pulled high by resistor **54**. The A**0** pin is driven low by the MMC card.

Detection logic in converter chip **40** reads card-select pins CD**0**, CD**1** to detect the presence of a flash-memory card. When a new card is present, detection logic then reads pins A**0**, A**1** as inputs. While A**0** is low, A**1** is high. The detection logic in converter chip **40** recognizes the LH state of A**0**, A**1** as indicating that a MMC or SD card is plugged into connector **44**. Converter chip **40** then exits detection mode and configures its interface to connector **44** for the 9-pin MMC/SD interface as shown later in FIG. **5**.

In FIG. **4**D, a SmartMedia card is inserted into the connector for card-type detection. SmartMedia card **24** (not shown) is plugged into SmartMedia adapter **30** which is plugged into connector **44**. The adapter **30** does not connect pins A**0**, A**1** from the CompactFlash interface to any pins on the SmartMedia card. Adapter **30** internally connects pin A**1** from the CompactFlash interface to the ground pin on the CompactFlash interface. The SmartMedia card does not drive either pin A**1**, A**0**, although adapter **30** drives pin A**1** low. Likewise, converter chip **40** does not drive pins A**1**, A**0** during detection mode. Pin A**0** floats and is pulled high by resistor **52**.

Detection logic in converter chip **40** reads card-select pins CD**0**, CD**1** to detect the presence of a flash-memory card. When a new card is present, detection logic then reads pins A**0**, A**1** as inputs. While A**0** is high, A**1** is low. The detection logic in converter chip **40** recognizes the HL state of A**0**, A**1** as indicating that a SmartMedia card is plugged into connector **44**. Converter chip **40** then exits detection mode and configures its interface to connector **44** for the 22-pin SmartMedia interface as shown later in FIG. **5**.

In FIG. **4**E, a Memory Stick card is inserted into the connector for card-type detection. Memory Stick card **18** (not shown) is plugged into Memory Stick adapter **34** which is plugged into connector **44**. Detection logic in converter chip **40** reads card-select pins CD**0**, CD**1** to detect the presence of a flash-memory card. When a new card is present, detection logic then reads pins A**0**, A**1** as inputs. Both pins A**0**, A**1** are low. The detection logic in converter chip **40** recognizes the LL state of A**0**, A**1** as indicating that a Memory Stick card is plugged into connector **44**.

Other form factors that are electrically compatible to SmartMedia, Secure Digital Card, MultiMediaCard, and/or Memory Stick may be detected in similar manner as described above with respect to SmartMedia, Secure Digital Card, MultiMediaCard, and/or Memory Stick. Some examples of the new form factors of SmartMedia include NAND MLC flash memory, NAND SLC flash memory, AG-AND flash memory, xD Card, etc. Some examples of the new form factors of Secure Digital Card include miniSD, SD HS, TransFlash, T-Flash, etc. Some examples of the new form factors of MultiMediaCard include RS MMC, MMC HS, etc. One example of the new form factors of Memory Stick is MS Duo.

FIG. **5** is a table of pin mappings for the SmartMedia, MMC/SD, and Memory Stick to CompactFlash adapters. The pin numbers for the smaller interfaces for SmartMedia, MMC/SD, and Memory Stick are not shown but can be in any order or designation. The adapter connects the proper pin on the smaller interface to the CompactFlash pin number shown in FIG. **5**. Simple wiring such as individual wires, flat cables, printed-circuit board (PCB), or wiring traces can be used.

The ground pins on the smaller interfaces are connected to CompactFlash pins **1** and **50**. Power pins are connected to CompactFlash pins **13**, **38**. Pins **25**, **26** are the card detect signals for CompactFlash, which the adapters connect to the card-detect signals on all smaller interfaces.

The CompactFlash connectors use pins **2-6**, **21-23**, **27-31**, and **47-49** for the 16-bit parallel data bus to the CompactFlash card. Pins **8**, **10-12**, and **14-20** form a separate 11-bit address bus. The separate data and address buses provide for rapid random addressing of CompactFlash cards. Other control signals include pins **6**, **32** chip enables, pin **9** output enable, pin **36** write enable, interrupt pin **37**, reset pin **41**, and register REG pin **44**. REG pin **44** is the Attribute Memory Select, defined based on the CF mode of operation, i.e. PCMCIA I/O mode, IDE or PCMCIA Memory Mode. Several pins in the 50-pin interface are not connected.

The smaller SmartMedia interface also has a parallel data bus of 8 bits. These are mapped to pins **2-6**, and **21-23** of the CompactFlash interface to match the CompactFlash D**0**:**7** signals. While no separate address bus is provided, address and data are multiplexed. Control signals for latch enables, write enable and protect, output enable, and ready handshake are among the control signals. Output enable—OE and write enable—WE are mapped to the same function pins **9**, **36** of the CompactFlash interface. The total number of pins in the SmartMedia interface is 22.

The Memory Stick and MMC/SD flash-memory-card interfaces are smaller still, since parallel data or address busses are not present. Instead, serial data transfers occur through serial data pin DIO, which is mapped to pin **19** (A**1**). Data is clocked in synchronization to clock SCLK on pin **18**. A command signal CMD or BS occupies pin **20** (A**0**). The MMC/SD and Memory Stick interfaces require only 6 pins plus power and ground.

Detection logic in converter chip **40** reads card-select pins CD**0**, CD**1** to detect the presence of a flash-memory card. When a new card is present, detection logic then reads pins A**0**, A**1** as inputs to determine the card type. The pull-up resistors of FIG. **4**A together with wiring inside the adapter and the card's behavior determines whether A**0**, A**1** are pulled low by the adapter or pulled high by the pull-up resistors.

FIG. **6** is a diagram of a multi-slot embodiment of the flash-card reader. While the single-slot embodiment of FIG. **3**B results in the smallest physical design, somewhat larger flash-card readers can be made that have separate slots for each type of flash-memory card, rather than a single slot. This negates the need for the adapters.

Four connectors are provided in flash reader **42**: a 50-pin CompactFlash connector **62** that fits CompactFlash card **16**, a 9 pin MMC/SD connector **64** that fits MultiMediaCard **28** or a Secure Digital card, a 22-pin SmartMedia connector **66** that fits SmartMedia card **24**, and a 10-pin Memory Stick connector **68** that fits Memory Stick **18**.

Each of the four connectors **62**, **64**, **66**, **68** route their signals to converter chip **40**. Converter chip **40** detects when a flash-memory card has been inserted into one of the connectors **62**, **64**, **66**, **68** and configures itself to read files from the inserted card using the pin interface of FIG. **5** corresponding to the card type.

Converter chip **40** executes various routines to perform handshaking with the flash-memory cards and accept data, either serially or in parallel. The data is buffered and then sent to the host PC **20** through USB connector **46**. Converter chip **40** generates the appropriate USB-interface signals to transfer the data to host PC **20**.

Having separate connectors **62**, **64**, **66**, **68** with separate slots in flash reader **42** allows for card-to-card transfers. For example, images or other files from Memory Stick **18** could be transferred to CompactFlash card **16** by converter chip **40** reading serial data from Memory Stick inserted into connector **68**, converting to parallel, and writing to connector **62** and CompactFlash card **16**. Each of the flash-memory cards in connectors **62**, **64**, **66**, **68** can be assigned a different drive letter by the operating system, such as e:, f:, g:, and h:.

In this embodiment, flash reader **42** is contained in an external housing that connects to host PC **20** through a USB cable. Of course, other cables and interfaces such as IEEE 1394 FireWire may be substituted.

In an alternative embodiment, the flash reader **42** may include connectors for other types of flash media, such as SmartMedia-compatible cards (e.g., NAND MLC flash, NAND SLC flash, AG-AND flash, xD Card, etc.), Secure Digital-compatible cards (e.g., miniSD, SD HS, TransFlash, T-Flash, etc.), MultiMediaCard-compatible cards (e.g., RS MMC, MMS HS, etc.), and Memory Stick-compatible card (e.g., MS Duo, etc.).

Field-Operable Stand-Alone Apparatus for Media Recovery and Regeneration

FIG. **9** is a diagram of a field-operable stand-alone apparatus **70** that accepts several formats of flash-memory cards and can retrieve image data from damaged flash media and regenerate/repair such media so that it may be operable without requiring the use of a host PC. This type of apparatus may be useful in many situations. For example, users of digital cameras may not always have their PC's nearby. While extra flash-memory cards can be purchased and swapped in the digital camera, these flash-memory cards are somewhat expensive, especially when many high-resolution images are captured. Furthermore, in situations such as when users are traveling and are away from their PC, the users may be limited by the capacity of the flash-memory cards.

In one embodiment, field-operable apparatus **70** includes four slots with corresponding connections: A 50-pin CompactFlash connector **62** fits CompactFlash card **16**, a 9-pin MMC/SD connector **64** fits MultiMediaCard **28** or a Secure Digital card, a 22-pin SmartMedia connector **66** fits SmartMedia card **24**, and a 10-pin Memory Stick connector **68** fits Memory Stick **18**. Each connector is operably-coupled to a controller **72**, as depicted by bus lines **63**, **65**, **67**, and **69**.

Controller **72** detects when a flash-memory card has been inserted into one of the connectors **62**, **64**, **66**, **68** by sensing card select lines CD**0**, CD**1** in the manner discussed above, and automatically configures itself to read files from the inserted card using the pin interface of FIG. **5** corresponding to the card type. Controller chip **72** also executes software depicted as software instance **74**, which includes various routines to perform handshaking with the flash-memory cards and accept data, either serially or in parallel. As described below, software instance **74** also includes modules and procedures and the like embodied as executable instructions that enable the apparatus to perform data recovery, regeneration, and repair operations on damaged flash media when executed by controller **72**.

In general, field-operable apparatus **70** may include one or more storage means on which data are stored, including a removable mass storage device **76**, random access memory (RAM) **78**, and flash memory **80**. For example, embodiments that include a removable mass storage device may use such a device for storing image data retrieved from a corrupted flash media device. In this instance, data are read from the damaged flash media device, buffered, and then sent to the removable mass storage device to be written in a format compatible with the device, all via control by controller **72** through execution of software instance **74**. In general, the removable mass storage device will store the data on a removable media **77**, such as a standard or a high-density floppy diskette, a tape, a writeable CD-R/W disk, or other proprietary media such as LS120 by Imation of Oakdale, Minn., or ZIP drive media by Iomega Corp. of Roy, Utah.

In general, RAM **78** may be used for temporarily storing data or executable code during run-time operations. For example, as described below, all or most of software instance **74** will be stored in some form of non-volatile storage on the device, such as read-only memory (ROM). Since RAM is much faster than ROM, the executable code will first be loaded from the ROM into the RAM, where it may be executed by controller **72**. In other instance, RAM **78** may be used as a block buffer or large buffer that is used to install blocks of data retrieved from the damaged flash media. As described below, this retrieved data can then be reformatted, and written back to the flash media to regenerate the media.

Flash memory **80** may be used for several purposes. In one embodiment, flash memory **80** may be used as a flash store for storing images, including both images retrieved from corrupted flash media, and images simply read from uncorrupted flash media. For example, a user of apparatus **70** may wish to store several flash media cards worth of digital photos taken during a trip. Rather than bring and keep track of dozens or more flash media cards, the user would only have to bring a single card, and possibly a backup in case the original card was lost or physically damaged. Data could then be transferred to the on-board flash media of apparatus **70** whenever the flash card became full. Another consideration is that flash media cards are generally proprietary, as discussed above. This typically results in flash media cards being significantly more expensive than the underlying flash memory chips used to store the card's data. As a result, an apparatus **70** capable of holding, e.g., a dozen Memory Stick cards worth of data might actually cost less than the dozen Memory Sticks, proving to be of value without even considering the apparatus' data recovery and regeneration capabilities.

As another option, apparatus **70** may provide a communications interface, such as a USB interface **46** to enable communication with another device, such as a desktop or laptop PC, a PDA, a pocket PC, etc., in a manner similar to that described above. This would further extend the inherent functionality of the apparatus. For instance, if a user employed flash memory **80** to store image data during a trip, the user could transfer such data to his PC when he returned home. As another example, field-operable apparatus **70** could be used for data recovery and regenerations features while in the field, while it could additionally be used as a universal flash media reader while in the home or office. Typically, the communication interface may be controlled via controller **72**, via either built-in control logic, firmware execution, or software execution, or a special purpose interface chip, such as a USB interface chip **40'** may be employed for such purposes.

In general, field operable apparatus **70** may be powered by one or many well-known power sources. For example, the apparatus may employ conventional replaceable batteries, such as AA- and AAA-size batteries, as depicted by battery symbol **82**. The apparatus may also include an external power input **84** via which a DC voltage may be externally supplied, such as through use of an AC/DC converter or through an automobile cigarette lighter. Optionally, external power input **84** may be used to supply external power to recharge one or more rechargeable batteries. When appropriate, additional power conditioning circuitry may be employed to ensure the apparatus electronics are provided with properly conditioned source voltages (not shown).

In the illustrated embodiment, field-operable apparatus **70** is provided with a simple user interface, including a light-emitting diode LED **86** and button **87**, along with an optional LCD display **88**. In one embodiment, data recovery/regeneration operations may be initiated via a sensed activation event. For instance, such an activation event might occur when the

user inserts a flash-memory card into one of connectors **62**, **64**, **66**, **68**. As another option, such an event might be generated in response to user activation of button **87**. In response, the apparatus determines which of connectors **62**, **64**, **66**, **68** has a memory card inserted, and performs one or more file recovery/media regeneration operations via execution of software instance **74**.

In one embodiment, LED **88** can be programmed to blink during the copying process, and remain lit when the copying is complete, or vice-versa. This provides a simple visual indication to the user of the copying progress. Errors can be indicated with additional LED indicator lamps, or other blinking arrangements or colors. Optionally, error conditions may be displayed on LCD **88**.

In an optional configuration, various information may be annunciated via a speaker **89** driven by an audio chip **90**. This could include both aural annunciations, as well as verbal annunciations that have been digitally recorded and stored in the apparatus' non-volatile memory. For example, the apparatus could annunciate "recovery completed," when a recovery operation was completed. Other information, such as number of files recovered, identification that a flash media card is physically damaged (i.e., unable to be repaired), etc., might also be verbally annunciated via speaker **90**. These operations could either be facilitated by controller **72** (through built-in functionality or execution of firmware or software) or through one or more dedicated components configured for such purposes (not shown).

In general, LCD **88** in combination with one or more buttons **88** (other buttons not shown) may be used to provide a simple user interface to enable users to select various operations; such as retrieval of files from damage media and repairing damaged media. For example, various options may be displayed on the LCD with corresponding indicia disposed adjacent to respective buttons, wherein activation of a particular button is used to select the listed option adjacent to the button.

A field-operable apparatus **70**A having an optional configuration in which the aforementioned SM-to-CF adapter **30**, MMC/SD-to-CF adaptor **32**, and Memory Stick-to-CF adaptor **34** may be employed along with a connector **44'** to enable the apparatus to interface with various types of media, including CompactFlash **16**, Memory Stick **18**, SmartMedia **24**, SD media **26**, and MMC media **28** is shown in FIG. **8**. In an alternative embodiment, the apparatus **70**A may include other adaptors to interface with other types of flash media, such as SmartMedia-compatible cards (e.g., NAND MLC flash, NAND SLC flash, AG-AND flash, xD Card, etc.), Secure Digital-compatible cards (e.g., miniSD, SD HS, TransFlash, T-Flash, etc.), MultiMediaCard-compatible cards (e.g., RS MMC, MMS HS, etc.), and Memory Stick-compatible card (e.g., MS PRO, MS ROM, MS Duo, etc.). In general, all of the components having the same reference numbers in FIGS. **7** and **8** perform substantially similar operations. In this instance, controller **72** is programmed to additionally perform the operations of converter chip **40** discussed above with reference to FIG. **3***b*. Optionally, a separate converter chip may be used for these operations.

FIG. **9** is a diagram showing high-level functional blocks corresponding to an exemplary controller **72**. Typically, controller **72** may be implemented as a commercially-available micro-controller chip that is programmed to read and write I/O pins that are connected to the flash-memory-card connectors and optionally USB interface. In the illustrated embodiment, controller **72** includes a CPU **91**, ROM **92**, On-board RAM **93**, timers **94**, a shifter **95**, and general purpose input/output (GPIO) **96**, each of which is connected to an internal bus **97**. Optionally, the controller may further include a built-in USB interface **98**.

Generally, various control and transfer routines will be written and stored in ROM **92** as executable instructions. At run time, these instructions are loaded from ROM **92** into RAM **93** and executed by CPU **91**. Included among these routines is a high-level scanning routine for sensing when a flash-memory card is inserted into one of connectors **62**, **64**, **66**, **68** or **44'**. Upon determining the particular type of flash media that is inserted, the instructions will be programmed such that CPU **91** will begin execution of one or more other routine specific to that type of flash-memory card, including transfer and handshake sub-routines.

General-purpose input-output **96** provide registers or I/O ports that drive external I/O pins of controller **72**, or read the logic-levels or voltages on input pins to controller (pins not shown). CPU **91** can read registers in GPIO **96** that are written by control signals that are coupled to I/O pins of controller from connectors **62**, **64**, **66**, **68** (or **44'** for apparatus **70**A). Control signals to the flash-memory cards can be switched high or low by writing a 1 or a 0 to a register for that control signal in GPIO **96**. GPIO also may be used for general I/O communication with control inputs (e.g., button **88**) and for control signals for annunciating LED **86**. GPIO can also be used for interfacing with removable mass storage device **76**. Optionally, a controller may be employed that has a built-in interface, such as an IDE interface, for providing this functionality.

Timers **94** are useful for asserting control signals for a required amount of time. For example, a control signal may need to be asserted for a specified number of microseconds. CPU **91** can write a 1 to a register in GPIO **96** and start a timer in timers **94**. For example, a designated timer can send an interrupt to CPU **91** when the specified time has elapsed, or CPU **91** can continuously or periodically poll timers **94** to determine when the specified time has elapsed. In response, CPU **91** can write a 0 to the register in GPIO **96**, causing the control signal to transition from 1 to 0.

Shifter **95** is connected to the data and clock signals from connectors **64**, **68**. When data is read from the flash-memory card, a clock is pulsed to synchronize the data transfer. Shifter **95** clocks in one bit (serial) or word (parallel) of data for each clock pulse. A cyclical-redundancy-check (CRC) can be performed on the data to detect errors. CPU **91** can request re-transmission of data from the flash-memory card when an error is detected.

Data read by shifter **95** can be sent over internal bus **97** to be stored in a buffer in RAM **92**, or optionally RAM **78**, depending on the size of RAM **92**. Later, CPU **91** can execute a routine to transfer this data from RAM **92** (or **78**) to removable mass storage device **76** or flash **80**. Subsequently, data stored in flash **80** (or even RAM **92** or **78**) may be transferred via USB interface **98** (or USB chip **40'**) to a computer linked to field-operable apparatus **70** with a USB cable connected to USB port **46**.

Software for Performing Data Recovery and Media Regeneration

In accordance with further aspects of the invention, details for implementing software for performing data recovery and flash media regeneration are now disclosed. Typically, executable instructions corresponding to programs, modules, routines, procedures, and the like will be stored in either ROM **92** and/or a separate persistent storage means, such as a separate ROM (not shown), or even flash **80** or other rewritable persistent storage means. For example, if flash **80** was

employed, software upgrades could simply be downloaded over the Internet to a PC, and then transferred to apparatus **70**, **70**A via, e.g., a USB link. As discussed above, at run time the instructions will generally be transferred to from the persistent storage means to RAM **92** (or **78**). In addition to the software discussed below, other executable instructions may include firmware (stored in ROM) for performing basic services, such as disk control/interfacing.

Generally, media cards may be accessed by computers readers or via built-in circuitry designed for reading (usually) a particular type of flash media, as discussed above. Typically, the reader or built-in circuitry, in combination with an appropriate operating system driver, provide an interface to a flash media device such that the device appears to the operating system (and thus application running on the operating system) as a disc storage device, such as a conventional hard disk drive, floppy drive, or CD-ROM drive. Accordingly, the storage structure of flash media devices is configured in a manner that supports a general disk access scheme. In support of this scheme, flash media often employ disk-like file system access schemes.

FIG. **10** shows a typical organization of a data stack in a flash card. At the bottom of stack **100** is the actual raw flash data **102**. The raw data may contain bad blocks, etc., which are usually dealt with on the file system level (i.e., FAT level) **104**. The main file system types are FAT**12**, FAT**16** and FAT**32**. All of these contain a master boot record (MBR) used in conjunction with a disk boot record (DBR) and file allocation table (FAT) to store computer-type. A flash header **103** generally contains information from which physical (i.e., raw data) to logical (i.e., file system level) mapping can be determined, such a Card Information Structure (CIS) information. Generally, the CIS contains information about the card manufacturer, many of the device characteristics, and is sometimes used to define manufacturer-specific features, such as whether data stored on the media correspond to digital pictures in a regular or panoramic, etc., also referred to as proprietary sub-formatting information for different camera types; and MP3 players, mostly using proprietary information. Similarly, devices that use Memory Stick media put down a header **103** known as a Boot Block to signify that the media has a known signature on it.

The header **103** also lets the system know what type of file system is used in layer **104**. The file system (FAT typically) contains information about which sectors form contiguous files, which sectors are bad and cannot be used, etc. In some cases they may also contain information about how often a sector has been used, to optimize rotation, but that type of information is not necessarily required, as the number of cycles performed by cameras is rather low when compared to typical PC applications.

On top of the FAT is the formatted data **105**, which is the data that is typically presented to a user. This data is often in a format such as Joint Photographic Expert Group (JPEG or JPG) files, Bitmap Picture (BMP) files by Microsoft, Graphic Interchange File (GIF) files by CompuServe, Motion Picture Expert Group Protocol Layer 3 (MP3) files, Waveform Audio File Format (WAV) files, Audio-Video Interchange (AVI) files, Tagged Image File Format (TIFF or TIF) files, Motion Picture Expert Group (MPG or MPEG) files, QuickTime Movie (MOV) files by Apple using QuickTime format, and Windows Media Audio format (WMA) files.

FIG. **11** shows a top-level flowchart illustrating operations performed by an embodiment of the novel software program described herein. The program initializes at a block **200**, which corresponds to the main entry point in the program. Typically, the initiation process will be invoked in response to a sensed activation event, such as insertion of a flash media card or user activation of button **88**. In a decision block **201** wherein a determination is made to whether the flash media comprises SmartMedia. If it does, the logic proceeds to a block **202**, which initiates the operations shown in FIG. **12***a*.

In accordance with a decision block **301** in FIG. **12***a*, the program looks at the SmartMedia flash header to determine if a CIS exists. Generally, the CIS will be stored in the first available physical block (e.g., physical block **0**, if this block is not damaged). As described above, the CIS includes information that is used for defining characteristics of the media, including its physical access structure (i.e., block size, number of blocks, etc.). For SmartMedia cards this (preferably) will correspond to the physical access structure defined in the SmartMedia Physical Format Specification. If no CIS is present, the program builds a Physical-to-Logical table through pass-through code, as described below with reference to FIG. **13**. Pass-through code is a code that allows the application to see the layers **101** through **104** (as shown in FIG. **1**) below the logical layer **105** that contains the file-formatted data. By looking at the data at the lowest levels, in particular at level **102**, the software can determine what was there and rebuild the sectors. Upon not finding a the CIS, a pass-through flag is marked as true in a block **302**, and a Pass-through Read ID command is issued to retrieve the SmartMedia size. The Physical-to-Logical table is then built using Pass-through commands in consideration of the SmartMedia size, as described below. Exemplary portions of the pass-through code application program interface (API) are provided in the Appendix section of this disclosure.

After the CIS is found, or the Physical to Logical table is built, the program proceeds to a block **305**, wherein, beginning with logical sector zero, as defined by the Physical-to-Logical table or identified by the CIS, the media is searched for the selected file types described above (e.g., JPEG, GIF, MPEG, etc). In one embodiment, the user may be prompted to indicate the file types present on the medium, or there may be an automatic lookup in a table (not shown) using a database (not shown) to determine which file types are supported. Details of this operation are continued at an entry point **520** in FIG. **14**.

Returning to FIG. **11**, if the flash media does not comprise SmartMedia, a determination is made in a decision block **203** to whether the media comprises a Sony Memory Stick (or Memory-Stick compatible media, e.g., MS Duo). If the answer is No, the program proceeds to a block **204**, which begins the operations discussed in the description below of FIG. **14**. If the program detects that the flash media is a Memory Stick, the program tests in a decision block **207** to determine whether there is a valid boot block. It is to be noted that Memory Stick media contain two Boot Blocks and if neither one is recognized, the user is prompted in a block **208** to give the capacity of the medium and then the program returns to the main flow, so that in either case (No or Yes), the program proceeds to a block **206**, jumping the flowchart description to FIG. **3***b*. In one embodiment, user interaction is enabled via verbal instructions issued via speaker **90**. For example, a verbal instruction could be issued such as, "push button once for 2 megabyte media, twice for 4 megabyte media, and three times for 8 megabyte media." In another embodiment, a small LCD screen, along with corresponding user input buttons, may be employed (not shown).

Referring now to FIG. **12***b*, in a decision block **309**, the program searches for a master boot record. If an MBR is not found, the pass-through flag is marked at TRUE in a block **310**, and a Physical-to-Logical table is built using Memory-Stick pass-through commands in accordance with a block **311**

and the operations of FIG. **13** described below. After finding the MBR or building the Physical-to-Logical table, the program proceeds to a block **512**, in which the program again starts from Logical Sector **0** (identified via either the MBR or the physical-to-logical table) and searches for the selected file types using a process similar to that performed in block **305** of FIG. **12***a* above. Details of this operation are continued at an entry point **520** in FIG. **14**.

Building the Physical-to-Logical Table

In general, a flash media device has a physical storage scheme corresponding to the lowest access level of the device. This is very inconvenient for software application and the like to use, since various flash media have different physical storage schemes. In order to provide a standard access interface, a file system interface (i.e., layer **104**) is provided. The file system handles the details of converting file access requests from an application (via an operating system call and using a corresponding OS driver) to a low-level request that is actually used to access the data. This is usually supported by a Physical-to-Logical mapping scheme, whereby the file system presents a logical representation of the storage device to the application. The file system layer also enables the file system to be standardized (e.g., FAT**12**, FAT**16**, FAT**32**, etc.), enabling access to the flash media using a well-known interface.

At it lowest level, flash media provide a physical storage scheme based on addressable units called blocks. A block-addressing scheme (vs. a pure linear addressing scheme) is used, in part, because individual bits on flash media devices are not re-writable. Accordingly, in order to rewrite data, one or more entire blocks must first be erased (i.e., reset), and then new data may be written to the block(s) by setting selected bits. For convenience, the physical addressing scheme may employ a page-based memory access scheme, wherein each block is further divided into one or more memory pages. This enables the device to support a page-by-page reading and (for blocks that have been reset) writing scheme, and a block-by-block erasing scheme. For example, an 8 Megabyte SmartMedia device has 1024 blocks, each comprising 16 pages of memory, wherein each memory page comprises 512 bytes plus 16 "extra" bytes, and wherein individual bytes may be accessed via a page address and an offset address.

As described above, the logical storage scheme for flash media is generally set up to mimic a disk storage device (i.e., hard drive), wherein the most basic storage unit comprises a sector which may map to one or more physical storage pages based on Physical-to-Logical mapping. The 16 extra bytes above are termed "extra" because in a corresponding disk device, each memory page will correspond to a 512 byte sector, and the sectors are contiguous without any extra data in between. In general, various types of schemes may be used to store the Physical-to-Logical mapping data, including redundant schemes. In SmartMedia and Memory Stick devices, logical block addressing information is stored in these "extra" bytes.

In view of the foregoing considerations, details for building the Physical-to-Logical Table are shown in the portion of the flowchart illustrated in FIG. **13**. As depicted by a block **404**, retrieval of the Physical-to-Logical mapping information begins by reading corresponding data in the extra byte portions of the blocks using appropriate Pass-through commands, beginning with physical block **0** and continuing to the end of the media. For each block, the following operations corresponding to blocks **406**, **408**, **410**, **412**, and **414** are performed, as appropriate.

First, in decision block **406**, a determination is made to whether the block is good. Such information can generally be obtained directly from the device using known methods. If the block is not good, the logic proceeds to a block **408**, which loops the logic back to begin processing the next block. If the block is good, the logic proceeds to a decision block **410** in which a determination is made to whether the block is a CIS block (e.g., for SmartMedia) or boot block (e.g., for Memory Stick). If the answer is Yes, the logic loops back via block **408** to beginning processing the next block. If the block is not a CIS or boot block, data identifying the Logical block number is extracted from the extra byte data for the block and corresponding Physical-to-Logical block mapping information is stored in a PHYSICAL_TO_LOGICAL linked list structure, wherein each entry maps a Physical block to its corresponding Logical block. In accordance with decision block **414** and block **408**, the operations of blocks **406**, **410**, **412**, are repeated until all of the blocks have been processed in a similar manner to complete building of the PHYSICAL_TO_LOGICAL linked list structure. This linked list structure comprises the Physical-to-Logical table.

Extracting File Data

Generally, the operations described with reference to the portion of the flowchart shown in FIG. **5** pertain to the extraction of file data from a corrupted flash media device, which may be accessed via block **204** from FIG. **11** or block **520** from FIGS. **12***a* and **12***b*. As discussed above, block **204** is encountered if the flash media is not a SmartMedia card, a Memory Stick, or any flash media electrically similar to the SmartMedia card or the Memory Stick. If such is the case, the first operation that is performed is to read Sector **0**, in accordance with a block **501**. Next, in a decision block **502** a determination is made to whether the MBR is valid. If the MBR is valid, the disc boot record (DBR) is read in a block **503** and tested for validity in a block **504**. If the DBR is valid, the starting sector is determined in block **505** to be from the Data Area. If the DBR is not valid, the starting sector to read is determined to be Logical Sector Zero, as depicted by a block **512**.

In case no valid MBR is detected in block **502**, the program branches to a block **510** to see if there is a valid DBR. If there is a valid DBR, it branches back to block **505**. If there is no valid DBR, the program proceeds to a block **511**, where the starting sector to read is determined to be Physical Sector **0**, and from block **511**, the program then branches back to block **521**

With continued reference to block **521**, this is the point at which the operations of FIG. **5** are joined from blocks **520** in FIGS. **12***a* and **12***b*, i.e., wherein the starting sector is Logical Sector **0**. Now all program processes are united at block **521**, where the program starts reading from the starting sector, whichever sector has been determined (Physical Sector **0**, Logical Sector **0**, or Data Area). The following operations are then performed to extract (i.e., read) data pertaining to previously inaccessible files, such as picture files. For example, due to corrupted flash media, such files may not be accessible due to FAT errors, file header errors, checksum errors, etc.

In accordance with a decision block **522**, if the PassThru flag is set (TRUE), the physical blocks are read using Pass-through commands in a block **524** by employing the PHYSICAL_TO_LOGICAL linked list created above. If the PassThru flag is FALSE, the reading of logical blocks is performed using normal operating system commands, as depicted by a block **526**.

As discussed above, the flash media is read searching for files corresponding to selected file types, which will typically

comprise one or more picture format files, such as JPEG, GIF, MPEG, BMP, etc. More specifically, in a block **528** a search is made of indicia via which one or more selected file types can be identified, which will typically comprise a file header, but may comprise other types of indicia. Generally, the file-type indicia may be identified by searching for a specific string or pattern. The following operations illustrate the use of a file header for the file indicia. Those skilled in the art will recognize that similar operations may be performed to extract file data through use of other file indicia as well.

Generally, the reading is performed in sequential bases, using either a sequence of physical block reads (mapped to corresponding logical sectors), or direct reading of a sequence of logical sectors via the operating system. In response to encountering file indicia, a determination is made in a decision block **530** to whether the indicia corresponds to a file header, and whether a corresponding free cluster is formatted. If both are true, the logic proceeds to a decision block **531**, wherein a determination is made to whether information identifying the file size (of the file) is included in the file header. If the file size is included in the header, the program continues to read data in accordance with a block **532** until data corresponding to the complete file has been read (as defined by the file size), whereupon the file is closed and the logic proceeds to a block **533** to read the next sector. (It is noted that a given file may span more than one sector.) In a corresponding decision block **540**, a determination is made to whether the previous sector is the last sector to be processed. If it is, the process ends in accordance with an end block **541**. If not, the program cycles back to block **530** to begin processing the next sector.

Returning to decision block **531**, if the program finds a header but does not find the file size, it proceeds to a block **535**, where it reads until the end of the current sector, and then returns to block **533**. This process is repeated until the last sector is reached as determined by decision block **540**, whereupon the program exits in accordance with end block **541**.

Those skilled in the art will recognize that the foregoing process is merely exemplary, and that the recover scheme may be modified in many ways. In some cases, for example, the high-level operations of the main program (FIG. **11**) may have many more branches, for different media types, or for proprietary file systems, etc., as are disclosed above.

In another embodiment the file allocation table (FAT) on the media is examined to determine its validity, and if it is valid, an attempt to retrieve the file system starting sector, size, etc., is performed. This method is useful if the image data is stored in non-contiguous clusters. A cluster is a minimum allocation unit that the operating system assigns to store data, and typically comprises one or more sectors. For instance, if a cluster is equal to four sectors and each sector is 512 bytes, then even a file of size 1 byte will take up 2048 bytes of space on the medium.

Regenerating Data on Corrupted Flash Media

Details of operations and logic performed when regenerating corrupted flash media in accordance with one embodiment are shown in the flowchart collectively shown in FIGS. **15-20**. With reference to the high level portion of the flowchart shown in FIG. **15**, the process begins with a decision block **600** in which a determination is made to whether the user has selected to transfer data from the corrupted flash media to internal memory or to external memory. In either instance, the data will be read from the flash media in the manner discussed above. In the case of a transfer to internal memory, the data are read and written to a persistent storage means onboard the field-operable apparatus, such as flash memory **80**, as depicted by a predefined process block **602**. In the case of a transfer to external memory, the use may place a flash media card in one of connectors **62**, **64**, **66**, **68**, or **44'** (as appropriate), and the data are written to the flash media card, as depicted by a predefined process block **604**. Upon writing to the internal or external memory, the process is complete.

If the answer to decision block **600** is no, the logic proceeds to a decision block **606** in which a determination is made to whether the media type is SmartMedia or any flash media electrically similar to SmartMedia (e.g., NAND MLC flash, NAND SLC flash, AG-AND flash, xD Card, etc.). If so, the logic jumps to the start of the flowchart portion shown in FIG. **16** at flowchart connector **1**. If not, a determination is next made in a decision block **608** to whether the media type is a Memory Stick or any flash media electrically similar to Memory Stick (e.g., MS Duo, etc.). If it is, the logic jumps to the portion of the flowchart shown in FIG. **17** at flowchart connector **2**. If the media type isn't Memory Stick, the logic flows to a decision block **610** in which a determination is made to whether the media type is MMC or any flash media electrically similar to MMC (e.g., RS MMC, MMC HS, etc.). If it is, the logic jumps to the portion of the flowchart shown in FIG. **18** at flowchart connector **3**. If it is not, the media is either SD, CF, or any flash media electrically similar to SD (e.g., miniSD, SD HS, TransFlash, T-Flash, etc.). In this instance, normal media checking is performed in a block **612**, completing the process. Generally, normal media checking comprises operations to verify whether the media is properly formatted, such as checking to see if both FAT**1** and FAT**2** are good, checking to see if there are multiple physical blocks mapped to identical logical blocks, etc.

With reference to flowchart connector **1** at the top of FIG. **16**, the first operation for SmartMedia media or any flash media electrically similar to SmartMedia comprises a determination of whether the media has a Card Information Structure (CIS) in a block **614**. If the answer is no, the logic proceeds to a block **616** in which the ID for the card is read and a corresponding CIS is created.

Whether the CIS exists or is created, the logic flows to a decision block **618** in which a determination is made to whether the master boot record (MBR) is OK. If it is, an MBR_OK flag is set in a block **620**. A determination is then made in a decision block **622** to whether the disk boot record (DBR) is OK. If it is, a DBR_OK flag is set in a block**624**.

Next, in a decision block **626** a determination is made to whether the root directory is OK. If it is, a ROOT_DIR_OK flag is set in a block **628**. Additional checks for media write-ability are then performed in a block **630**. Following this, a determination is made in a decision block **632** to whether the MBR_OK flag is set. If not, the MBR is restored per the media size using a predefined process **634**, further details of which are discussed below with reference to the flowchart portion shown in FIG. **20** at flowchart connector **8**. In a decision block **636** a determination is made to whether the DBR_OK flag is set. If not, the DBR, FAT**1**, and FAT**2** file allocation table instances are restored using a predefined process **638** in accordance with the Table **1** parameters. The logic then jumps to the flowchart portion shown in FIG. **19** a flowchart connector **6**.

Continuing at flowchart connector **6**, in a decision block **640**, a determination is made to whether the ROOT_DIR_OK flag is set. If not, the ROOT DIR area is restored and updated as new files are recovered from the media, as depicted by a block **642**. General file recovery operations are then performed in a block **644** in the manner discussed above. Upon recovering the files, the ROOT DIR area and the FAT**1** and

FAT2 areas of the media are updated accordingly. This completes the regeneration process for the SmartMedia.

If the media type is a Memory Stick or any flash media electrically similar to Memory Stick, the logic will jump to flowchart connector **2** on FIG. **17**. In a decision block **644**, a determination is made to whether Boot Block**1** is valid. If it is not, the logic flows to a decision block **646** wherein a determination is made to whether Boot Block **2** is valid. If neither Boot Block is valid, an error condition exists and the media cannot be recovered, as indicated by a block **648**. If either Boot Block**1** or Boot Bock**2** is valid, the logic proceeds to the portion of the flowchart shown in FIG. **16** at flowchart connector **7**. In this instance, the same operations and logic in blocks **618-644** will be performed in the manner discussed above.

With reference to FIG. **18**, if the media type is MMC or any flash media electrically similar to MMC, the first operation is to initialize the card in a block **650**. This involves reading the CSD to find the card size and write protecting the group size. Next, in a block **652**, a Send_Write_Protect command is sent to the entire card to see if any write protected groups exist. In accordance with a decision block **654**, if any write protected groups are found, the write protect bits for those groups are cleared using the Clear_Write_Port command in a block **656**. Once these operations are completed, the logic jumps to FIG. **16** at flowchart connector **7**, and the operation of blocks **618-644** are again performed in view of the media type being MMC.

Returning to FIG. **20**, the restoration of the MBR operations begin in a decision block **658** in which a determination is made to whether the media type is SmartMedia. If it is, the operations of blocks **660** and **662** are performed, wherein the MBR is rebuilt using data from Table 1 of FIG. **21** and locating an erased block and writing the MBR to that block. If the media type is not SmartMedia, a determination is made in a decision block **664** to whether it is Memory Stick. If it is, the MBR is restored in accordance with blocks **666** and **662** using the parameters shown in Table 2 of FIG. **22**. If the answer to both of decision blocks **658** and **664** are no, the media type is MMC, and the MBR is restored in a block **668** using MMC-specific parameters (not shown).

The above description of illustrated embodiments of the invention, including what is described in the Abstract, is not intended to be exhaustive or to limit the invention to the precise forms disclosed. While specific embodiments of, and examples for, the invention are described herein for illustrative purposes, various equivalent modifications are possible within the scope of the invention, as those skilled in the relevant art will recognize.

These modifications can be made to the invention in light of the above detailed description. The terms used in the following claims should not be construed to limit the invention to the specific embodiments disclosed in the specification and the claims. Rather, the scope of the invention is to be determined entirely by the following claims, which are to be construed in accordance with established doctrines of claim interpretation.

What is claimed is:

1. An apparatus comprising:

a housing means for housing at least a controller;

at least one flash media connector means for accepting at least one type of flash media;

storage means coupled to the controller;

means for detecting an activation event;

means for recovering data, in response to the activation event, the data corresponding to at least one application file from a corrupted flash media device coupled to the flash media connector means;

means for searching for file indicia corresponding at least one predetermined file type;

means for reading data based on information in the file indicia, said data comprising recovered data; and

means for storing the recovered data on the storage means.

2. The apparatus of claim **1**, wherein the flash media connector means is to accept at least one type of flash media either directly or via a flash media connector adaptor.

3. The apparatus of claim **1**, wherein the means for detecting an activation event is to recover the data by accessing the flash media at a low level to access raw flash data stored on the flash media device.

4. The apparatus of claim **1**, wherein the means for searching for file indicia comprises searching raw flash data stored on the flash media device for the file indicia.

5. The apparatus of claim **1**, wherein the means for reading data based on the file indicia, comprises reading data from raw flash data stored on the flash media device.

6. The apparatus of claim **1**, wherein the flash media connector means comprises means for determining a type of media connected to the media connector means.

7. The apparatus of claim **6**, wherein the means for determining a type of media connected to the media connector means comprises means for determining the type of media by reading one or more address pins on the media.

8. The apparatus of claim **1**, further comprising control and transfer means for performing data transfers and handshake routines based on a type flash media detected to be connected to the flash media connector means.

9. The apparatus of claim **1**, wherein the flash media comprises one of a CompactFlash compatible card, a SmartMedia compatible card, a MultiMediaCard compatible card, a Secure Digital compatible card, a Memory Stick compatible card, and a Universal Serial Bus (USB) drive.

10. The apparatus of claim **1**, wherein the activation event comprises detecting a flash media to be connected to the flash media connector means.

11. The apparatus of claim **1**, wherein the activation event comprises a user activation of a button on the apparatus.

12. An apparatus comprising:

a flash media connector means for accepting at least one of a plurality of different flash media;

a means for reading and writing to a flash media coupled to the flash media connector means;

a means for recovering data from a corrupt flash media coupled to the flash media connector means, in response to detecting an activation event; and

a means for storing recovered data.

13. The apparatus of claim **12**, wherein the flash media connector means is to accept at least one type of flash media either directly or via a flash media connector adaptor.

14. The apparatus of claim **12**, wherein the means for recovering data comprises means for searching for file indicia corresponding to at least one predetermined file type, and reading data based on information in the file indicia.

15. The apparatus of claim **14**, wherein the means for reading data based on the file indicia, comprises reading data from the raw flash data.

16. The apparatus of claim **12**, wherein the flash media connector means comprises means for determining a type of media connected to the media connector means.

17. The apparatus of claim **16**, wherein the means for determining a type of media connected to the media connector means comprises means for determining the type of media by reading one or more address pins on the media.

18. The apparatus of claim **12**, further comprising control and transfer means for performing data transfers and hand-

shake routines based on a type flash media detected to be connected to the flash media connector means.

19. The apparatus of claim **12**, wherein the flash media comprises one of a CompactFlash compatible card, a SmartMedia compatible card, a MultiMediaCard compatible card, a Secure Digital compatible card, a Memory Stick compatible card, and a Universal Serial Bus (USB) drive.

20. The apparatus of claim **12**, wherein the activation event is in response to detecting a flash media to be connected to the flash media connector means.

21. The apparatus of claim **12**, wherein the activation event comprises a user activation of a button on the apparatus.

22. A method comprising:

a flash media connector accepting at least one type of flash media;

detecting an activation event;

recovering data, in response to the activation event, the data corresponding to at least one application file from a corrupted flash media device coupled to the flash media connector;

the recovering data comprising searching for file indicia corresponding to at least one predetermined file type, reading data based on information in the file indicia, said data comprising recovered data, and storing the recovered data on the storage unit.

23. The method of claim **22**, wherein the flash media connector accepting further comprises accepting at least one type of flash media either directly or via a flash media connector adaptor.

24. A method comprising:

accepting, by a flash media connector, at least one of a plurality of different flash media;

reading a flash media coupled to the flash media connector;

recovering data from a corrupt flash media coupled to the flash media connector, in response to detecting an activation event; and

storing recovered data.

25. The method of claim **24**, wherein the flash media connector accepting further comprises accepting at least one type of flash media either directly or via a flash media connector adaptor.

26. An apparatus comprising:

a connector configured to accept two or more different types of flash media;

a controller configured to control reading from a corrupted flash media coupled to the connector, the flash media including data to be recovered;

the controller further configured to recover data from the corrupted flash media; and

a storage device coupled to store the recovered data.

27. The apparatus of claim **26**, wherein the storage device is a flash memory.

28. The apparatus of claim **26**, further comprising:

recovering data from an uncorrupted flash media; and

storing the data recovered from the uncorrupted flash media in the storage device.

29. The apparatus of claim **26**, wherein the controller is further configured to:

search the flash media for file indicia corresponding to at least one predetermined file type; and

read data from the flash media based on information in the file indicia.

30. The apparatus of claim **29**, wherein the reading the data based on the file indicia comprises reading data from raw flash data.

31. The apparatus of claim **26**, wherein the connector is configured to determine a type of media connected to the connector by reading one or more address pins on the media.

32. The apparatus of claim **26**, wherein the controller is further configured to perform data transfers and handshake routines based on a type of flash media detected as being connected to the connector.

* * * * *